

sinclair

נסרק על ידי

אונדסקן



ZX Spectrum +

מדריך למשתמש





## תוכנת ספקטרום

כל מגוון התוכנה עבור מחשבי ספקטרום (כולל כל  
הנושאים הקיימים כיום) תואם בשלמותו למחשב  
החדש ספקטרום ZX+.

## גאים להציג את הספקטרום ZX+

חברת סינקלייר ריסרץ' המובילה בטכנולוגיית המחשב הזעיר הביאה את  
המחשב לכל בית. בעקבות פיתוחו של המחשב הזעיר הראשון בעולם  
שהיה שווה לכל כיס, דגם ZX80, הושגו יכולת חישוב גדולה עוד יותר  
ותכונות משופרות נוספות – במחשבי ZX81, ספקטרום ZX ו-ZX QL. מילת  
המפתח היתה תמיד הנוחות בשימוש, גם בתכנון המוצרים וגם בדרך  
פעולתם.

הספקטרום ZX+ מציע את סינקלייר ריסרץ' צעד נוסף בדרך זאת.  
בתוכו, תוכל למצוא מכונה המשלבת את כל מעלות הספקטרום בגירסה  
משופרת ההופכת מחשב פופולרי ורב-עצמה זה לנוח בשימוש עוד יותר  
מקודמו. תקוותנו היא שתוכל לנצל את מלוא האפשרויות שמעניק  
המחשב החדש שלך

Chris Smith



# תוכן העניינים

---

---

## ההתחלה 3

---

## צעדים ראשונים בתכנות 17

---

## למד על הספקטרום $ZX+$ 41

---

## למד על הבייסיק של סינקלייר 49

נכתב ע"י ניל אודלי  
הוצא לאור ע"י דורלינג קינדורסלי בע"מ  
בשיתוף עם  
סינקלייר ריסרץ' בע"מ



# כיצד להשתמש בספר זה

מדריך זה עבור מחשב ספקטרום ZX+ מכיל 4 פרקים בקידוד צבעוני. כדי לפנות לפרק מסוים, פשוט פתח את הספר בחלק עם הצבע הנכון.

## 1 ההתחלה

הכנת הספקטרום ZX+ לפעולה ■ כוונן מקלט הטלביזיה ■ איתור תקלות הכנה לפעולה ■ מה מסוגל הספקטרום ZX+ לעשות ■ כיצד להשתמש בתכנה מוכנה לריצה ■ כיצד לטעון תכנית ■ איתור תקלות טעינת תכנה.

## 2 צעדים ראשונים בתכנות

המקלדת – לוח הבקרה של המחשב ■ כיצד להפעיל את המקשים ■ מחשבון הטלביזיה ■ הצבע וכיצד להשתמש בו ■ גרפיקת עשה זאת בעצמך ■ לוח טיוטה על המסך ■ תכנן בעצמך צורות ותמונות שונות ■ כיצד ליצור תווים משלך במחשב ■ הנפשה ■ כיצד ליצור מוסיקה ואפקטים קוליים ■ כיצד לשמור את תכניותיך ■ מאתר תקלות איחסון תכנה.

## 3 למד על הספקטרום ZX+

מה נמצא בפנים? ■ כיצד פועל הספקטרום ZX+? ■ כיצד לחבר רכיבים היקפיים ■ מיפוי זכרון בספקטרום ZX+.

## 4 למד על הבייסיק של סינקלייר

מדריך עיון למילות מפתח עבור מתכנת בייסיק של סינקלייר ■ דוחות מסך בספקטרום ■ מה מעבר לביסיק ■ מילון מונחי מחשב



# ההתחלה

פרק זה יורה לך כיצד להתחיל ולגלות את יכולתו של הספקטרום ZX+ כאן תמצא כיצד להכין את המחשב לפעולה כך שייגיב מיידית על דרישותיך, ואז בידך הברירה. אתה יכול להקיש פנימה מספר תכניות אשר יצעידו את הספקטרום דרך מרבית פעולותיו ויציגו את מלוא יכולתו בגרפיקה ואפקטים קוליים, או שתוכל ללמוד כיצד להשתמש בתכנה מוכנה לריצה, כגון משחקי מחשב. כך או כך, אתה תיהנה במהרה מהמחשב החדש שברשותך.



# הכנת הספקטרום ZX+ לפעולה

כדי להכין את הספקטרום לפעולה, עבור תחילה על רשימת התיוג שלהלן כדי להבטיח שיש ברשותך את כל החלקים הדרושים. אחר כך פעל לפי ההוראות בעמוד שממול לחבורים ולהתחברות לרשת החשמל. בשלב זה אינך זקוק לרשמקול.

**רשימת תיוג: האם יש ברשותך כל מה שנדרש?**

- 1 מחשב ספקטרום ZX+
- 2 ספק כח ZX — מייצר
- אספקת 9 וולט ז"י הנדרש ע"י הספקטרום.
- 3 כבל אנטנה — מחבר את הספקטרום למקלט הטלביזיה.
- 4 כבל קלטת — מחבר את הספקטרום לרשמקול.
- 5 תעודת אחריות — עליך למלא אותה ולהחזירה לסוכן או ליצרן.
- 6 קלטת המכילה מדריך למשתמש.
- 7 ספר הדרכה זה.
- מה שאתה נדרש לספק:
- 1 מקלט טלביזיה
- 2 רשמקול
- 3 תקע לרשת החשמל הביתית.



## שאלות ותשובות לשלב ההכנה לפעולה

**האם אני חייב להשתמש במקלט טלביזיה צבעוני?**  
לא. ברם, לא תוכל לראות את הצבעים שמפיק הספקטרום במכשיר שחור-לבן.

**האם כל מקלט טלביזיה מתאים?**  
הספקטרום אמור לתת תמונה עם כל מקלט טלביזיה שברשותך. במידה ולא, יתכן שמקלט הטלביזיה והמחשב משתמשים במערכות צבע שונות. הדבר עלול לקרות אם מקלט הטלביזיה ישן מאוד, או אם המקלט והמחשב נרכשו במדינות שונות. אם אתה בספק, פנה ליצרן מקלט הטלביזיה.

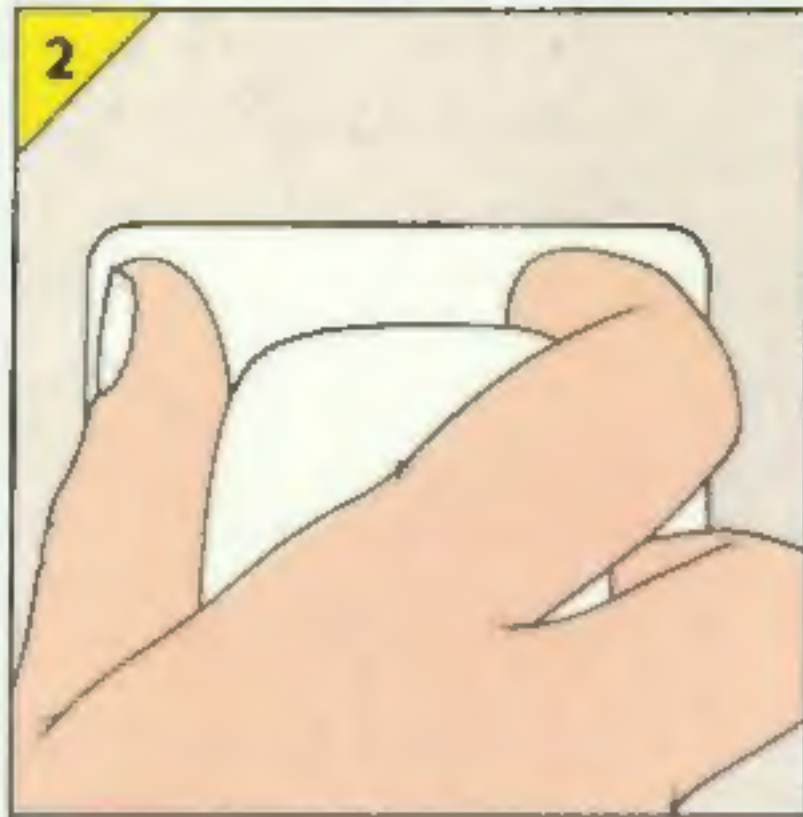
**האם אני יכול להשתמש בצג במקום מקלט טלביזיה?**  
כן. ניתן להשיג צגים המפיקים תמונה באיכות מעולה עם הספקטרום.

**איזה תקע דחש לרשת החשמל?**  
הספקטרום צורך 1.4A במתח של 240V/50HZ, מתח הרשת התיקני בבריטניה.

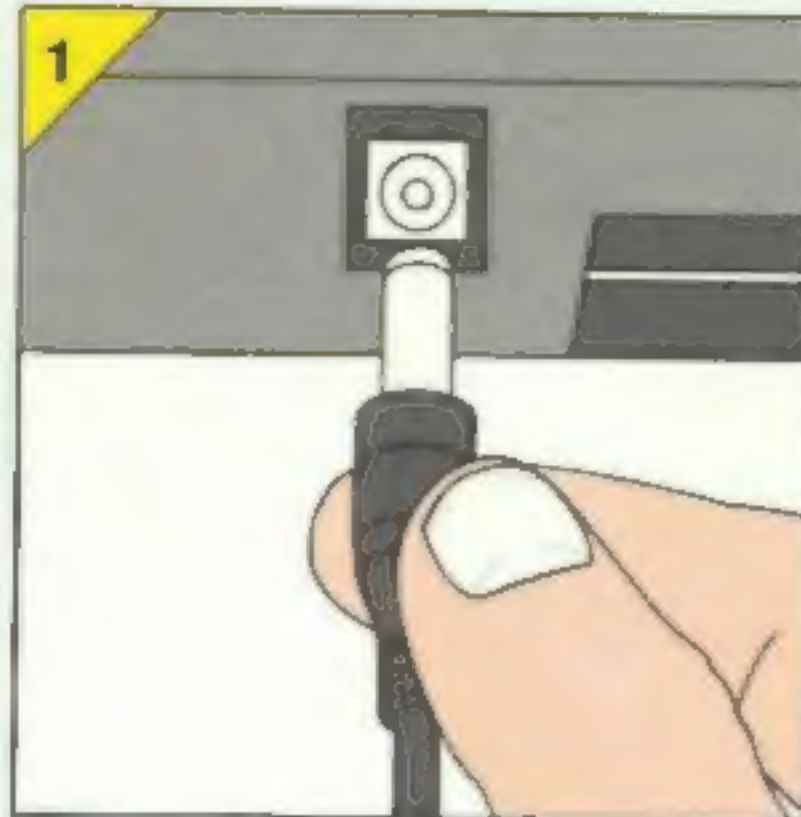
**האם הספקטרום מייצר הפרעות?**  
הספקטרום עלול להפריע לרדיו הנמצא בקרבת המחשב. לא ייגרם כל נזק לרדיו או למחשב.

**האם אני יכול להשתמש בראם ZX 16?**  
לא. מחסנית ראם זו מיועדת לשימוש במחשב סינקלייר ZX81 בלבד.



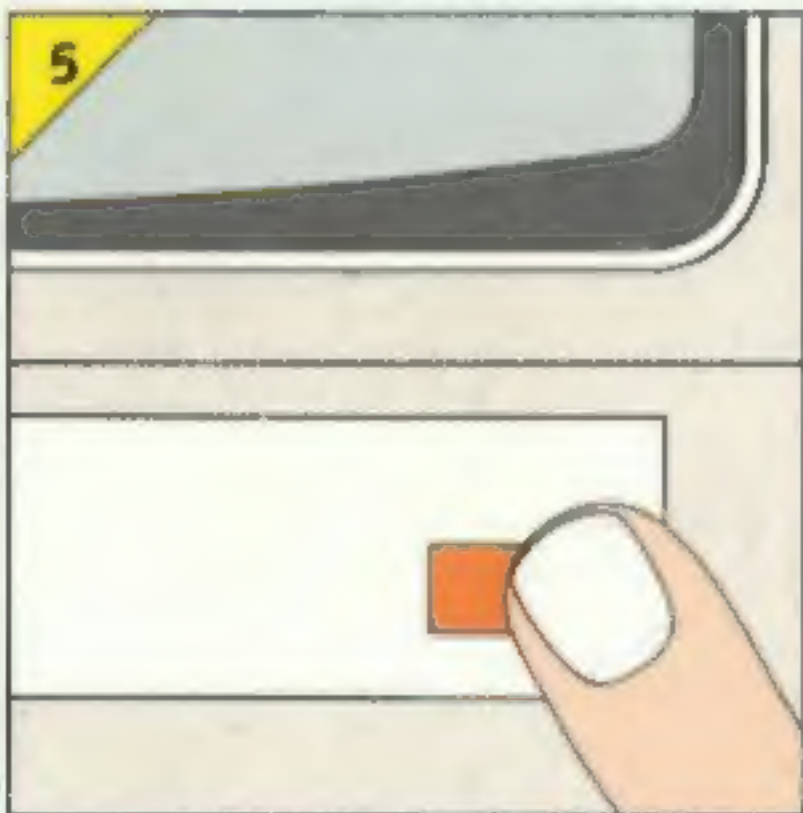


הכנס את התקע הקטן שעל כבל ספק הכח לשקע המסומן 9VDC בספקטרום

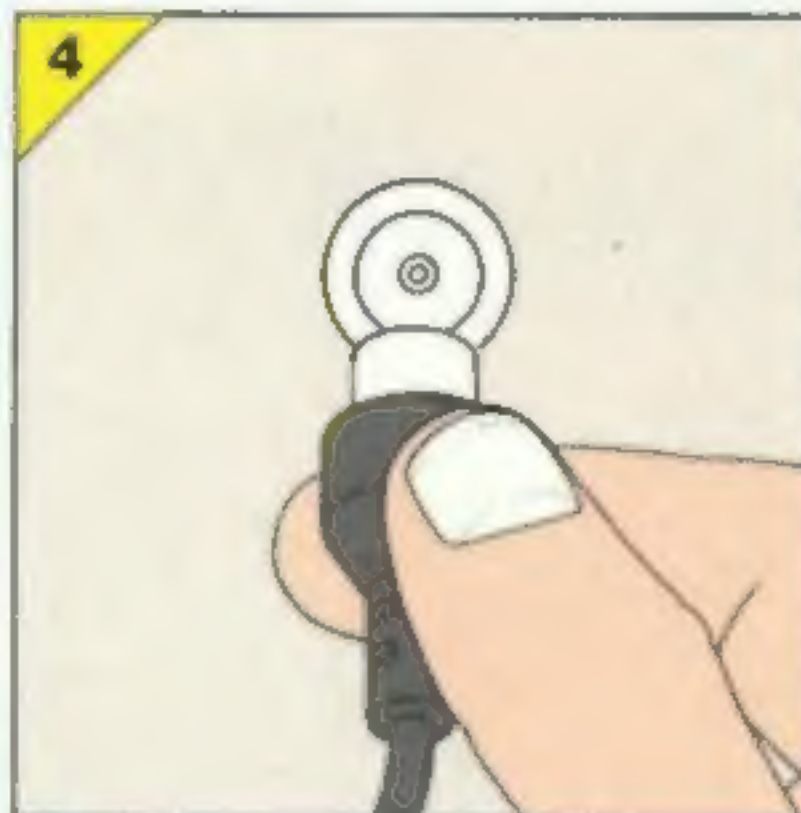


הכנס את התקע לשקע רשת החשמל והפעל אותו אם יש עליו מפסק. לספקטרום אין מפסק הפעלה משלו.

התחל בהתאמת תקע חשמלי לשני הקצוות החשופים של החוטים מספק הכח. כמו כן עליך להתאים נתיך 3A עבור התקע. שים לב לכך שהספקטרום אינו דורש חיבור הארקה לאדמה על אף שיתכן ולתקע שברשותך יש 3 פינים. אחר כך עקוב אחר סדר הפעולות המאוייד לחיבור הספקטרום לרשת החשמל הביתית ולמקלט הטלביזיה שלך. עם השלמת כל החיבורים והפעלת המערכת, עבור לעמוד הבא כדי ללמוד על כוונן מקלט הטלביזיה.



הפעל את מקלט הטלביזיה וסובב מייד את פקד עוצמת הקול כלפי מטה. אתה מוכן עתה לשלב כוונן המקלט כך שיקלוט את אותות הספקטרום.



נח את כבל האנטנה ממקלט הטלביזיה שלך. הכנס את התקע השני בכבל האנטנה של הספקטרום לשקע האנטנה במקלט הטלביזיה.



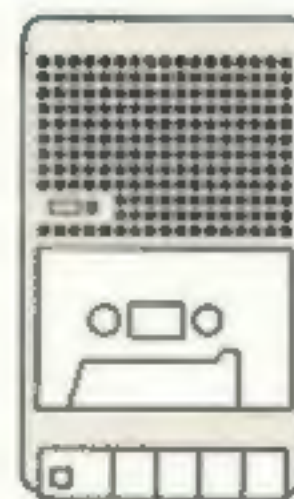
הכנס את כבל האנטנה לשקע המסומן TV בספקטרום. רק אחד מהתקעים בכבל האנטנה יתאים לשקע זה.

### שקעים ומחברים בספקטרום

שקע כח (POWER) – אספקת 9 וולט די המיוצרת על ספק הכח ZX. השנאי מחובר דרך שקע זה.



מחבר קצה – ניתן לחבר כאן מגוון רחב של חסרה, כולל כוונן זיזי מדפסת, ומודם.



שקע מיקרופון (MIC) – כאן מתחבר שקע המקרופון של רשמקול קלטות במטרה לשמור (להקליט) תוכניות על קלטות.

שקע אזנייה (EAR) – שקע האזנייה של רשמקול קלטות מתחבר לכאן במטרה לטעון בספקטרום תכניות שהוקלטו על קלטות.

שקע טלביזיה (TV) – כאן מתחבר שקע האנטנה של מקלט טלביזיה כך שניתן יהיה לראות את תמונות הספקטרום על המסך.





# כוונון מקלט הטלביזיה

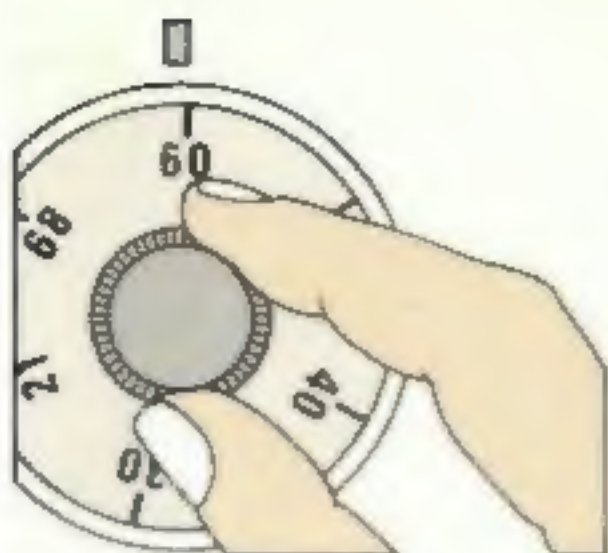
מחשב הספקטרום מפיק אות ווידאו לטלביזיה צבעונית בתדר המתאים לערוץ 36 בתחום התדרים UHF, כך שמקלט הטלביזיה חייב להיות מכוון לערוץ זה כדי לקלוט את תמונות המחשב.

לאחר הפעלת הספקטרום וחיבורו למקלט הטלביזיה, כוונן את פקד הכוונון במקלט עד שתקבל את הודעת הספקטרום כמתואר במסך הראשון להלן. לאחר שתראה זאת, אתה מוכן לבדיקת צבעי הספקטרום והתחלת התכנות. אם אינך מקבל את הודעת 'זכויות יוצרים', או אם הצבעים אינם נראים באיכות טובה, עיין בטבלת איתור תקלות בעמוד הבא.

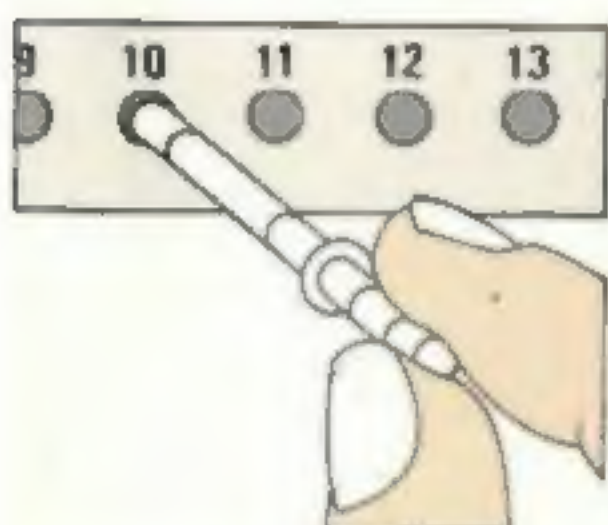
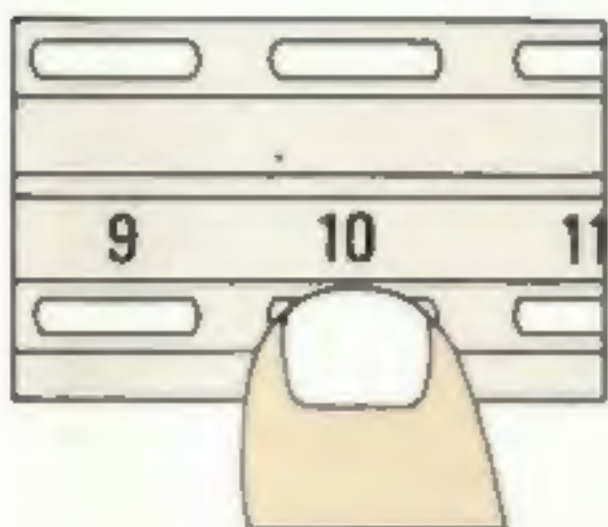


## פקדי כוונון

כוונון משתנה – פקד כוונון משתנה בורר כל ערוץ. פשוט סובב את הכפתור עד לקבלת הודעת הספקטרום.



כוונון בלחצנים – בחר כפתור כוונון עבור המחשב וכוונן אותו עד לקבלת הודעת המחשב. במידת האפשר, השתמש בכפתור פנוי, כך לא תצטרך לכוונן את המקלט בכל פעם שהמחשב מופעל.



כוונון אלקטרוני – במערכת זאת, המקלט עצמו מתכוונן לערוץ הנדרש. מקלטי טלביזיה עם ברירת ערוצים סינתטית ללא אפשרות כוונון ידנית עלולים להיות לא מתאימים לשימוש עם מחשב זה.



## כיצד לבדוק את צבעי הספקטרום

כדי לבדוק את צבעי הספקטרום, פשוט לחץ על מקש 8 ואחר כך על מספר מ'1 עד 6. הודעת 'זכויות יוצרים' תיעלם; ראשית תופיע המילה BORDER ואחריה המספר. כעת לחץ על מקש ENTER, איזור ה'גבול' שבמסך ישתנה לצבע המסומן במקש המספר. המסכים שלהלן מתארים מה קורה

לאחר הקשת 4 BORDER, ואח"כ 3 BORDER, ואת הקשת 7 BORDER תחזיר את איזור הגבול לרקע לבן.

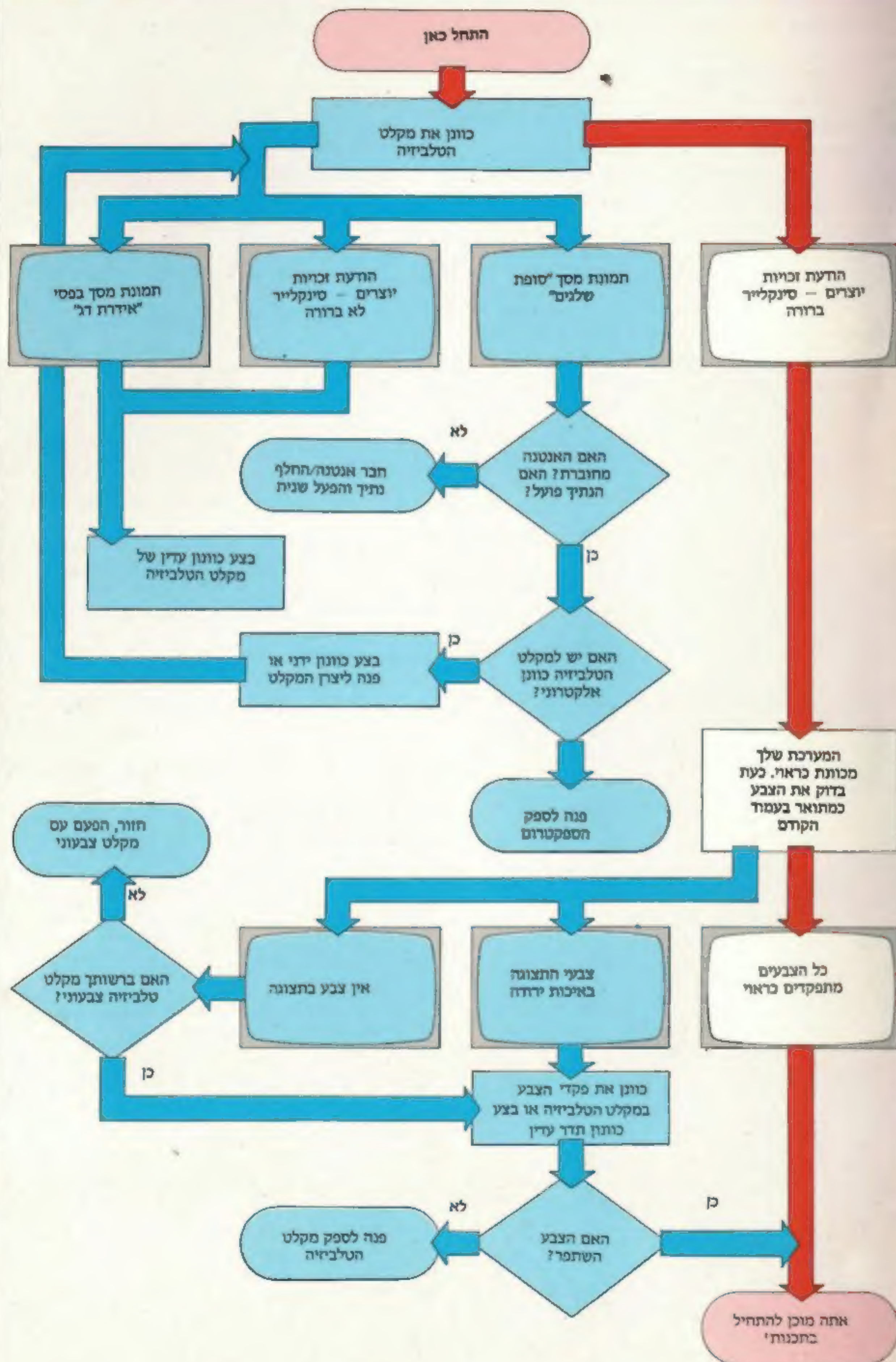
### BORDER 4



### BORDER 3









## כעת, תכנת את הספקטרום

הספקטרום מסוגל לבצע דברים רבים. אך כדי שהוא יפעל, עליך לספק לו מערך פקודות הנקרא תכנית מחשב.

להלן מופיע אוסף תכניות קצרות שיציעו את הספקטרום דרך שלבי פעולה שונים. כל שעליך לעשות הוא להקיש את פקודות התכנית בדיוק כפי שהן מופיעות כאן. תמונות המסכים מתארות למה עליך לצפות, אך אם תקרא את הפיסקה 'כיצד לשנות תכנית' בעמוד הבא, תוכל לבצע בעצמך ניסויים בתכניות.

## כיצד להקיש ולהריץ תכנית

כל מערך פקודות מתואר ברשימה הנקראת רישום (LISTING). אתה תראה שרישומי התכניות מכילים מספר חלקים שכל אחד מהם מתחיל במספר – 10, 20, וכן הלאה. כל חלק נקרא שורה בתכנית (אפילו אם הוא תופס שתי שורות במסך), ומכיל פקודה אחת או יותר למחשב. בכל שורת תכנית, תראה מילים שלמות או קיצורים המכילים שתי אותיות או יותר, כגון GO TO, PAPER, PI, RND, LET, PRINT.

## שמות

```
10 BORDER 1: INK RND*7
20 PAPER RND*7
30 PRINT "ZX Spectrum +";
40 GO TO 10
```



השם ספקטרום ZX+ מופיע בצבעים רבים על פני כל המסך. המחשב עוצר ומופיעה הודעת SCROLL בתחתית המסך. כדי לאפשר לתצוגה 'גלגול' כלפי מעלה, לחץ על מקש כלשהו מלבד N, SPACE, BREAK או STOP. אם אתה עוצר את הגלגול, ולוחץ BREAK, ואח"כ R(RUN) מלווה ב-ENTER. השמות יופיעו בצורות צבעים שונות.

## נסה זאת

בשורה 30, שנה "Zx Spectrum+" לשמך במרכאות ("") – למשל

```
30 PRINT "John";
```

זכור לכלול נקודה ופסיק (;). שמך יופיע מפורד על גבי כל המסך.

# מה מסוגל הספקטרום ZX+ לעשות?

## ראשית, ניסיון

כעת, לאחר שהפעלת את הספקטרום ומקלט הטלביזיה מכוונן כראוי, נסה ללחוץ על מספר מקשים. על המסך יופיעו מילים ואותיות, ויתכן שגם מספרים.

ברם, אם אינך יודע כיצד לתכנת את הספקטרום, קרוב לוודאי שהמחשב כלל לא יגיב להקשותיך. אך אל דאגה – שום נזק לא יכול להגרם, ולא חשוב על אילו מקשים לחצת.

כעת לחץ על מקש RESET בצד שמאל של המחשב, ואתה מוכן להפעלת הספקטרום. ארבעת העמודים הבאים יציגו בפניך מספר הדגמות על מסך הטלביזיה, שיראו מה הספקטרום מסוגל לעשות.

## כיצד להקיש

כדי להקיש מילה כלשהי, אות או מספר, שים לב תחילה למצבה על המקש. פעל לפי סדר הפעלת מקשי הברירה המתואר להלן.

מילת מפתח עליונה  
לחץ EXTEND MODE  
ואח"כ על המקש.

מילת מפתח תחתונה או  
סימן – לחץ  
EXTEND MODE, החזק  
מקש SYMBOL SHIFT  
לחוץ, ולחץ על המקש  
הדרוש.



מילת מפתח עליונה  
(חלק מורם) – לחץ על  
המקש.

אות או מספר (חלק  
מורם) – לחץ על  
המקש הנדרש. החזק  
מקש CAPS SHIFT לקבל  
אותיות גדולות.

מילת מפתח תחתונה או  
סימן (חלק מורם) –  
החזק מקש  
SYMBOL SHIFT לחוץ  
ולחץ על המקש הנדרש.

ניתן למצוא פרטים מלאים על הפעלת המקשים בעמ' 20-21.



המתן עד לגמר התכנית או הקש BREAK. אולי לחץ V(CLS) ואז ENTER מלווה ב(LIST) K. רישום התכנית (רשימת השורות) תופיע על המסך.



זהה איזו שורה אתה רוצה לשנות, ואז הקש את כל השורה החדשה, כולל מספר השורה, ולחץ ENTER. השורה החדשה תחליף את מקום הישנה. לחץ (RUN) R או ENTER, והתכנית תחדש תרוע.

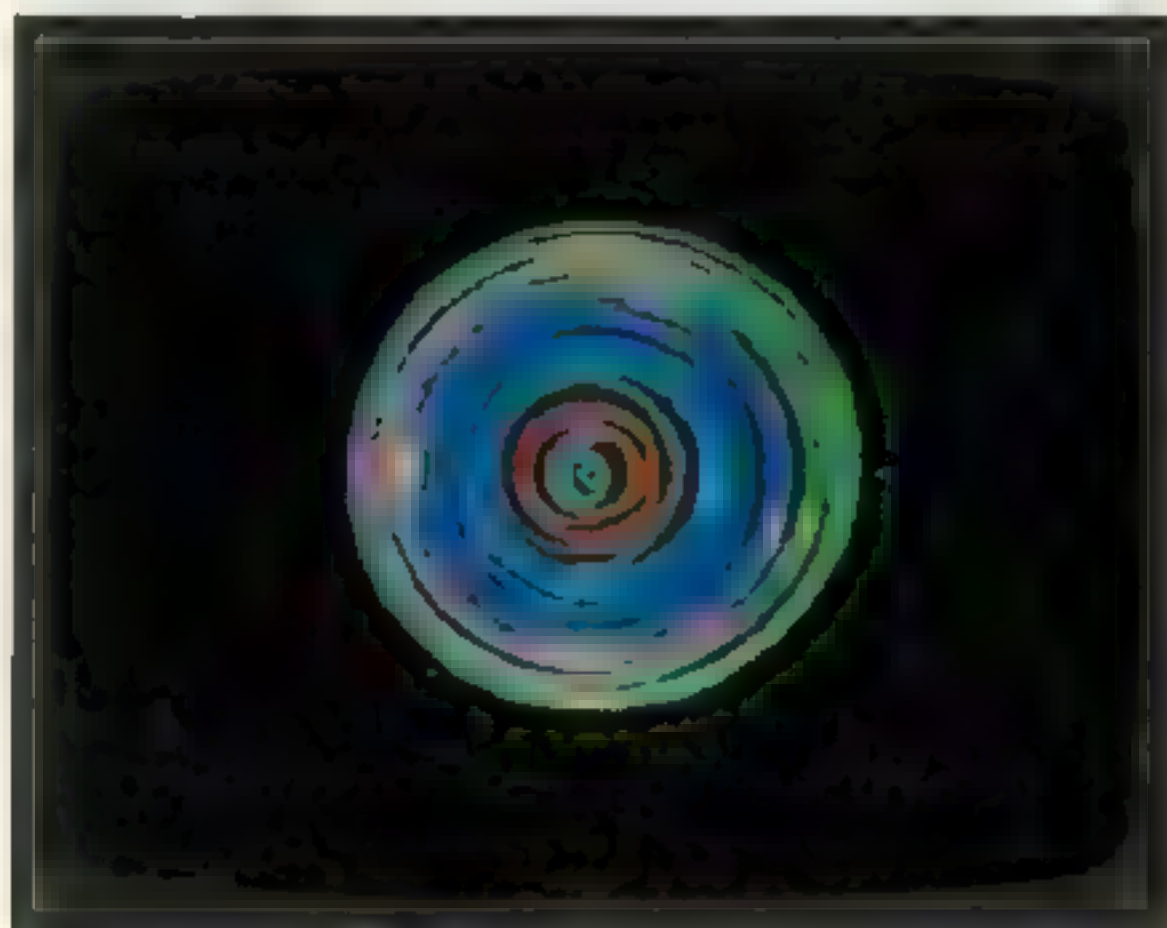
אלה נקראות מילות מפתח ואינך חייב להקישן אות אחרי אות. במקום זאת, מצא את המקש במקלדת אשר נושא את מילת המפתח (לדוגמה, PRINT נמצאת על מקש P), ומלא אחר ההוראות המפורטות בפיסקה 'כיצד להקיש'.

כאשר אתה מקיש שורה, היא מופיעה בתחתית המסך. כאשר אתה מגיע לסוף שורת התכנית, לחץ על מקש ENTER. השורה תופיע כעת בראש המסך. כעת הקש והכנס כל שורה באותה צורה. אם תקיש בטעות מקש לא נכון, עיין בפיסקה 'כיצד לתקן טעויות' בעמוד הבא.

כאשר סיימת להקיש את כל השורות, לחץ R. מילת המפתח RUN תופיע. כעת לחץ ENTER והספקטרום יזנק לפעולה עם הרצת התכנית.

### עיגולים מבזיקים

```
10 BORDER 0: PAPER 0: CLS
20 CIRCLE INK RND*6: FLASH RND
: 120+RND*5, 50+RND*5, RND*50
30 BEEP 0.1, RND*50
40 IF RND>.9 THEN GO TO 50
50 GO TO 20
60 FOR Y=-2 TO 4
70 FOR X=0 TO 6
80 BORDER X
90 BEEP .05, X+Y
100 NEXT X
110 NEXT Y
120 RUN
```



על המסך נבנה מערך של עיגולים מבזיקים כמעט קונצנטריים במגוון צבעים שונים. ואז, בבת אחת הגבול מבזיק, המחשב מפיק צליל שריקה, ומערך עיגולים חדש מופיע.

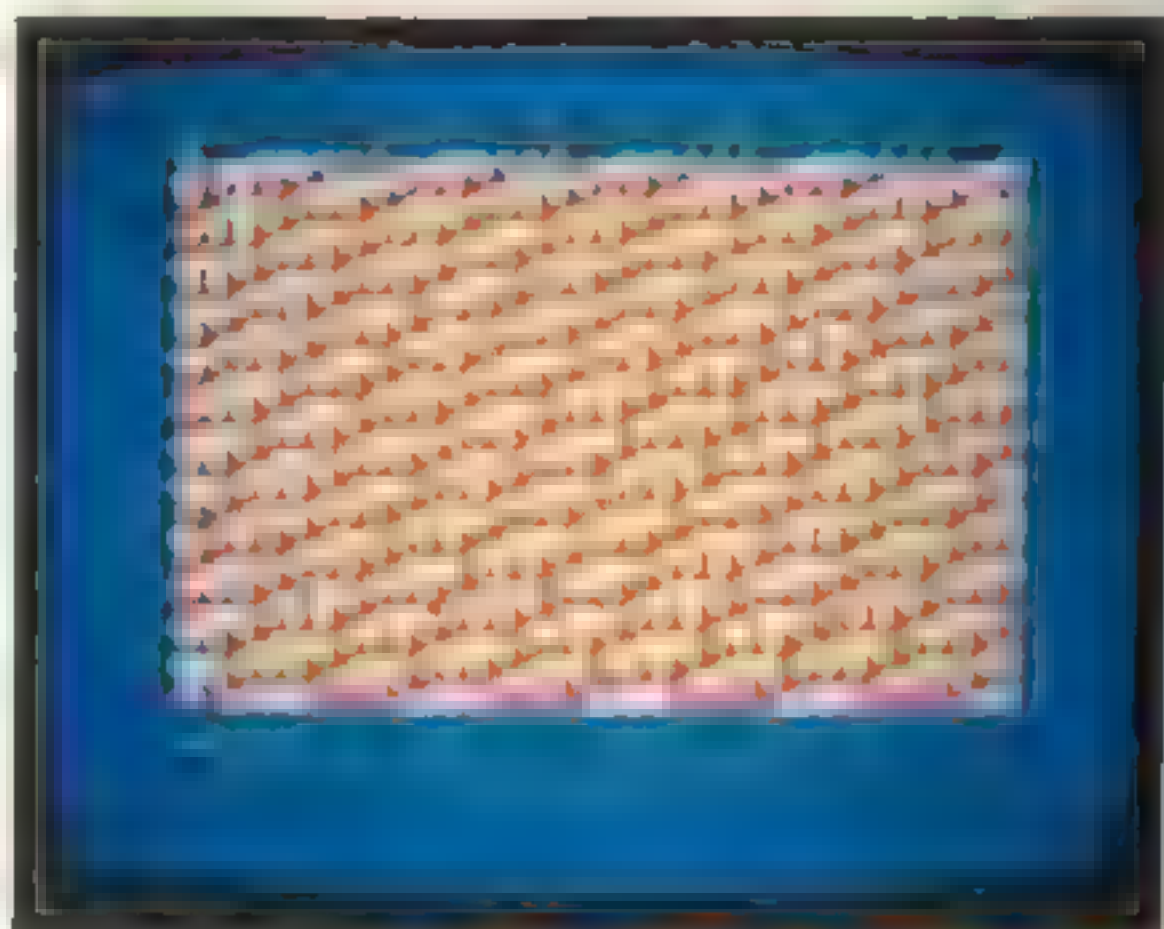
#### נסה זאת

לפני רישום התכנית (בעזרת מקש A), הקש פנימה PAPER 7 ולחץ ENTER.

אולי הקש שנית את שורה 20, הפעם בהשמטת שתי מילות המפתח FLASH RND, ואז העיגולים לא יבזיקו עוד.

### צורות

```
10 LET A$=""
20 FOR X=1 TO 7
30 LET A$=A$+CHR$(RND*14+129)
40 NEXT X
50 INK RND*7
60 BORDER RND*7
70 PRINT A$
80 GO TO 20
```



עם הרצת התכנית, תופיע על המסך צורה גיאומטרית צבעונית. כאשר המסך מלא, התצוגה תעצור בהודעת SCROLL כדי לראות יותר מתצוגה זאת, הקש משהו (מלבד N, SPACE, BREAK או STOP) כדי לגרום לצורה לעלות כלפי מעלה. כדי לראות צורה חדשה בצירוף צבעים אחר, הקש N עם הופעת ההודעת SCROLL. אולי הקש BREAK, מלווה ב(RUN) R או ENTER.

#### נסה זאת

בשורה 20, שנה 7 למספר אחר כדי לקבל צורה אחרת. נסה 8, לדוגמה.



## מוזאיקה מטורפת

```

5 BORDER 0 CLS
10 LET h=16 LET v=11
20 LET x=INT (RAND*3-1) : LET y=
  INT (RAND*3-1)
30 INK RAND+7
40 FOR z=1 TO 20
50 PRINT AT v,h,CHR$ 143
60 LET h=h+x
70 LET v=v+y
80 IF h<0 THEN LET h=31
90 IF h>31 THEN LET h=0
100 IF v<0 THEN LET v=21
110 IF v>21 THEN LET v=0
120 NEXT z
130 GO TO 20

```



ריבוע צבעוני מתרוצץ הלך ושוב על פני כל המסך, ובונה צורה צבעונית. בכל פעם שתפעיל מחדש את התכנית, תיבנה צורה חדשה.

נסה זאת

בשורה 50, שנה 143 ל-42 ואז תראה כוכבים! נסה מספרים אחרים בתחום 33 עד 142. עיין בטבלת מערך התווים לבדיקת התווים הצפויים.

## כיצד לתקן טעויות

אם אתה לוחץ על מקש לא נכון או לא לוחץ בנדרש על מקשי SHIFT או EXTEND MODE, אל תדאג. לוחץ על מקש DELETE, ומילת המפתח, סימן, אות או מספר, האחרון שהוקש יעלם. המשך ללחוץ על DELETE כדי למחוק תווים נוספים.



אם בצעת טעות לאחר לחיצת ENTER, עלול להופיע סימן שאלה מהבהב לפני השגיאה. לוחץ DELETE למחיקת השורה עד מקום השגיאה וכולל השגיאה, ואז השלם את השורה כראוי ולחץ ENTER. אם הצלחת להכניס שורה לא נכונה, התכנית עלולה להפסיק לפעול, ולהפיק דוח בתחתית המסך המציין את מספר השורה השגויה. הקש את השורה המתוקנת במלואה, ולחץ ENTER, R(RUN) או ENTER כעת התכנית אמורה לפעול.

## פוליהדרה

```

5 BORDER 1 PAPER 6 CLS
10 INPUT n
20 FOR r=20 TO 80 STEP 2
30 LET x=128 LET y=87
35 LET h1=x-r LET v1=y
40 PLOT h1,v1
50 FOR a=0 TO 361 STEP 360/n
60 LET h2=x-r+COS (a*PI/180)
70 LET v2=y-r+SIN (a*PI/180)
80 DRAW h2-v1,v2-v1
90 LET h1=h2 LET v1=v2
100 NEXT a

```



בתחילה, אתה רואה מסך ריק. הקש 6, ולחץ ENTER צורה בת 6 צלעות תיבנה על המסך, התחל שנית את התכנית כאשר הסתיימה והקש מספר אחד לקבלת צורה עם מספר צלעות שונה.

נסה זאת

בשורה 20, שנה 2 למספר אחר. הצורה נבנית מהר יותר אם המספר גדול יותר, והפוליהדרה (צורות בעלות משטחים רבים) מרוחקות יותר זו מזו.

## כיצד להתחיל מחדש תכנית

כמה מתכניות אלה – כמו 'פסיס וכוכבים' – מסתיימות ומפיקות את הדוח OK 0, עם מספר השורה האחרונה בתכנית. הדוח מציין שכל התכנית הסתיימה. כדי להתחיל מחדש, פשוט לחץ R(RUN) או ENTER. תכניות אחרות ממשיכות לרוץ, כמו 'מוזאיקה מטורפת', או מתחילות שנית באופן אוטומטי, כמו 'זריחה מנצנצת'.



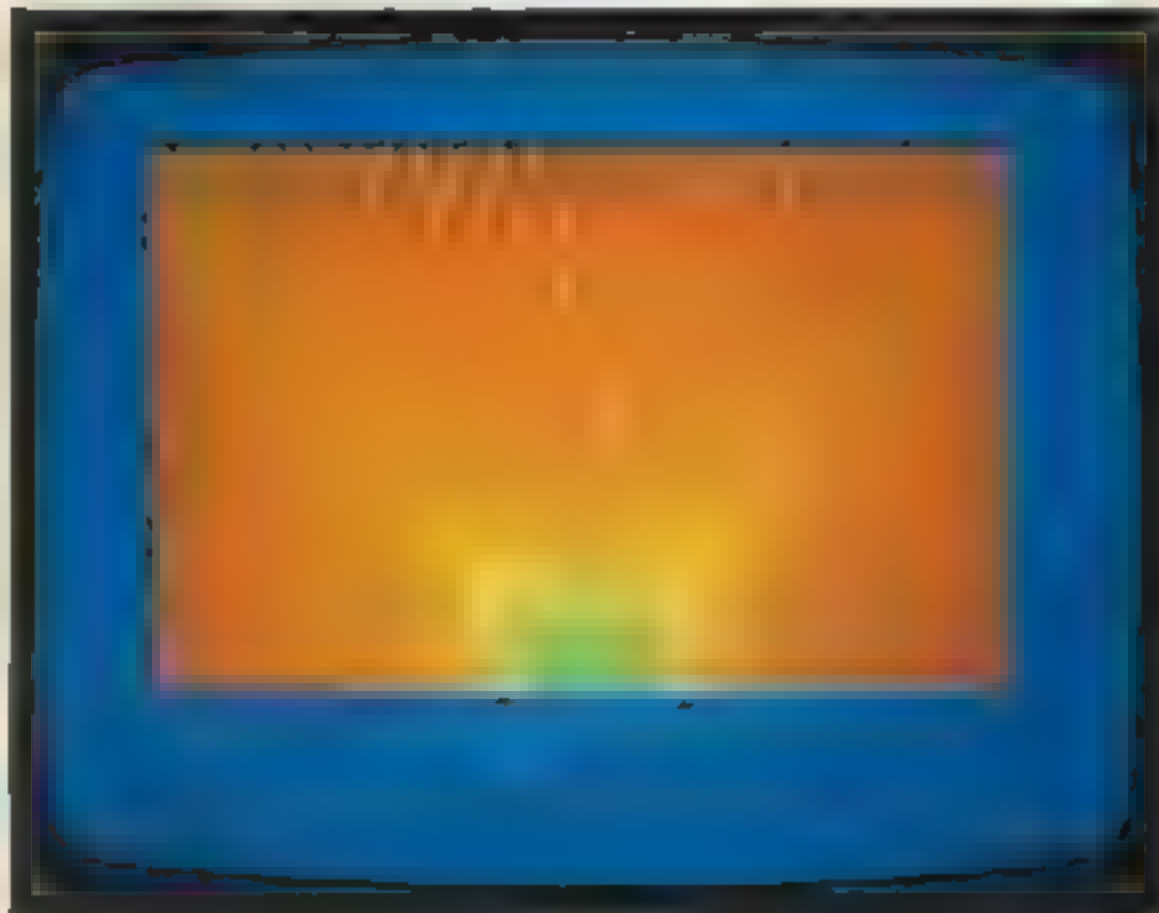
כדי לעצור תכניות כאלה, לחץ BREAK המשך ללחוץ על מקש זה עד שהתכנית נעצרת ומופיע דיוור BREAK על מנת להתחיל מחדש, פשוט לחץ R(RUN) או ENTER.



```

10 BORDER AND#6
20 INK AND#7
30 PAPER AND#6
40 CLS
50 LET Z=RND*10+2
60 FOR X=0 TO 174 STEP Z
70 PLOT 128,0
80 DRAW -128,X
90 BEEP .01,X/3
100 NEXT X
110 FOR X=-127 TO 127 STEP Z
120 PLOT 128,0
130 DRAW X,175
140 BEEP .01,60
150 NEXT X
160 FOR X=174 TO 0 STEP -Z
170 PLOT 128,0
180 DRAW 127,X
190 BEEP .01,X/3
200 NEXT X
210 PAUSE 200
220 GO TO 10

```



תמונה הדומה לזריחה מנוצצת בשלל צבעים נבנית מחדש בכל כמה שניות. אם המסך מתרוקן, פשוט המתן. בקרוב תראה שחר חדש מפציע.

נסה זאת  
כשורה 210, שנה 200 למספר אחר במטרה לשנות את פרק הזמן בו נשארת כל זריחה על המסך. 200 שווה ל-4 שניות.

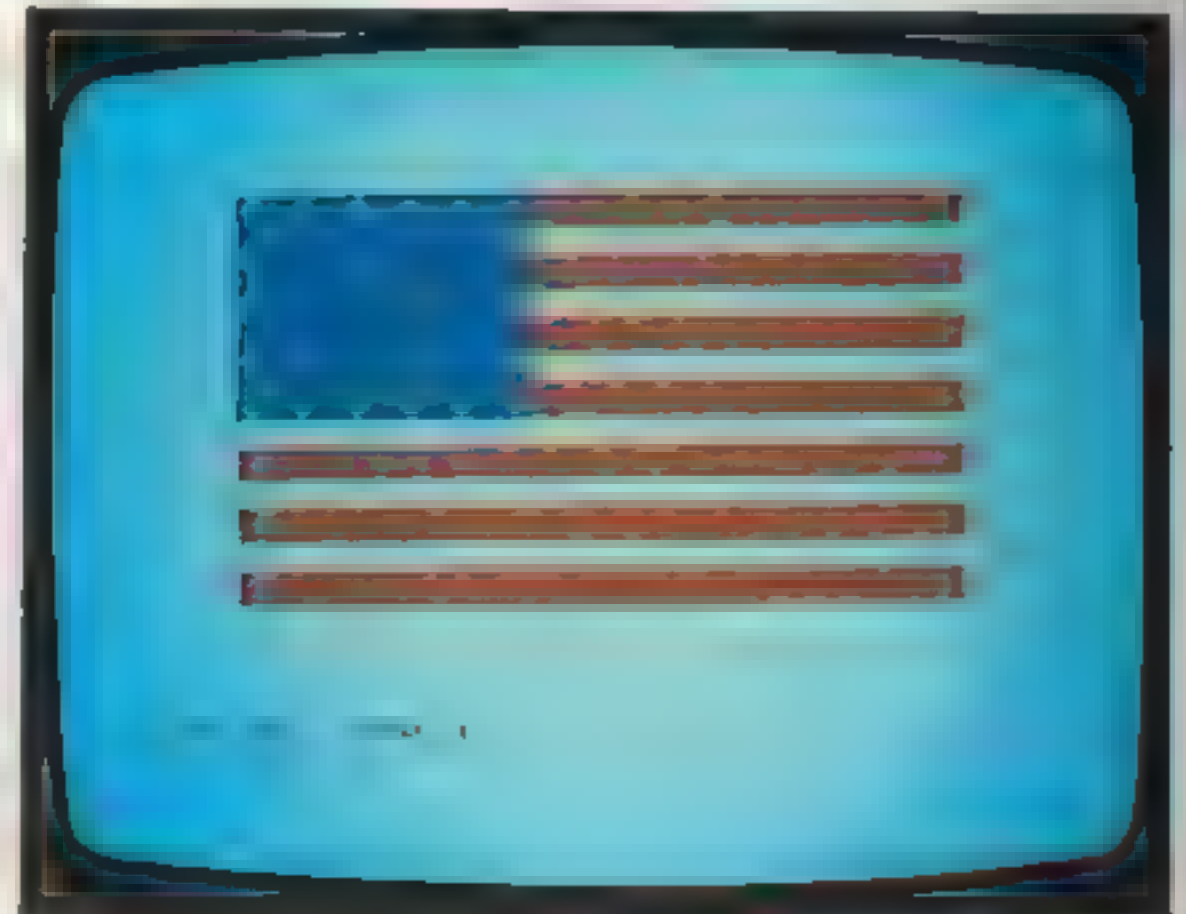
#### מה הלאה?

כעת יש לך ברירה. אם ברצונך לשמור חלק מתכניות אלה כדי להריצן שנית בעתיד, אתה יכול להקליטן על קלטות. עיין בפיסקה 'כיצד תשמור את תכניותיך' בעמ' 38 כדי ללמוד כיצד עושים זאת. אם ברצונך להמשיך בניסויים עם הספקטרום, תוכל ללמוד על תכנות בפרק 2 – צעדים ראשונים בתכנות. עד עתה רק ניסית תכניות כתובות, בלי להבין בהכרח כיצד הן פועלות. פרק 2 יסביר כמה מתכנות התכנות של הספקטרום. אם ברצונך לנסות כמה סרטי תכנה, כמו משחק מחשב שקנית, עיין בפיסקה הבאה 'כיצד להשתמש בתכנה מוכנה לריצה'.

```

10 INK 2
20 PAPER 7
30 CLS
40 FOR X=28 TO 148 STEP 20
50 FOR Z=0 TO 11
60 PLOT 16,Z+X DRAW 216,0
70 NEXT Z
80 NEXT X
90 PLOT 16,28 DRAW 0,131
100 PLOT 232,28 DRAW 0,131
110 PAPER 1
120 INK 7
130 FOR X=2 TO 8 STEP 2
140 PRINT AT X,2,"* * * * *"
150 PRINT AT X+1,2,"* * * * *"
160 NEXT X
170 PRINT AT X,2,"* * * * *"

```

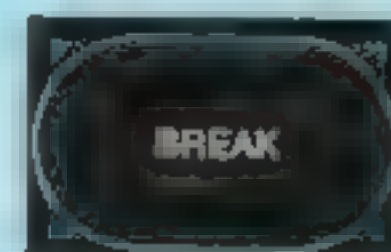


על המסך יופיע דגל ארה"ב.

נסה זאת  
שנה את מספרי הצבע בדגל. צבע הפסים נמצא בשורה 10, הכוכבים בשורה 120 והרקע לכוכבים בשורה 110.

#### כיצד להתחיל תכנית חדשה

כאשר סיימת לעבוד עם תכנית מסוימת וברצונך להכנס לתכנית חדשה לגמרי, המתן עד סיום התכנית הנוכחית או עצור אותה בלחיצת BREAK במצב זה תוכל לבחור באחת משתי דרכים למחיקת התכנית הישנה מזכרון המחשב. דרך אחת היא ללחוץ על שני מקשים, (A/NEW) ואז ENTER המסך ישחיד לדג, ואז תופיע הודעת 'זכויות יוצרים'.



לחלופין, ובצורה קלה יותר, תוכל פשוט ללחוץ על כפתור RESET לכפתור זה יש אותה השפעה כמו כיבוי הספקטרום על ניתוקו מאספקת הזרם וחיבור שנית.



# כיצד להשתמש בתכנה מוכנה לריצה

כאשר הכנסת תכנית לספקטרום, יצרת רצף של אותות מקודדים בצורה אלקטרונית כשלחצת על המקשים. הקודים מופנים לזכרון הספקטרום, אשר אוגר אותם כדי שהמחשב יוכל להשתמש בהם כאשר התכנית רצה. הקודים נשארים בזכרון עד שאתה מסיר אותם (ע"י הקשת NEW או לחיצת כפתור RESET, לדוגמה) או מכבה את הספקטרום. ברם, אין זה תמיד הכרחי להקיש פנימה תכנית כאשר ברצונך להשתמש בספקטרום. במקום זאת, אתה יכול לרכוש תכנה מוכנה לריצה, המכילה תכניות שניתן להזין ישירות למחשב באופן אוטומטי. השימוש בתכנה מוכנה לא רק חוסך לך את זמן ההקשה בכל פעם שתפעיל את הספקטרום, אלא מאפשר גם בניית ספריית תכניות מוכנות לשימוש, אשר היו דורשות ימים ואף שבועות של עבודה אילו היית כותב אותן בעצמך. יצרני תכנה מייצרים תכניות מכל הסוגים, אשר נכתבות בידי מתכנתים מעולים, ועבור הספקטרום מוצע מגוון עצום של תכניות. עיין בקטלוג התכנה של ספקטרום כדי לקבל מושג באילו תכניות תוכל להשתמש. ואז ברגע שתרצה, תוכל להריץ תכנית המתאימה לצרכיך.

## כיצד לטעון תכניות

אותות הקוד על סרט תכנה מכילים סדרת צפצופים גבוהים ונמוכים המוקלטים בקצב של כ-1500 בשנייה. כאשר אתה משמיע סרט תכנה ברשמקול, המכשיר מייצר את רצף הצפצופים המהווים את התכנית. כל שעליך לעשות הוא לחבר את הרשמקול לספקטרום, והקודים יעברו ישירות לזכרון המחשב. תהליך זה נקרא טעינת תכנית. בפרק זה תלמד כיצד לחבר את רשמקול הקלטות. עמ' 14-15 מתארים כיצד להשתמש בו.

## שאלות ותשובות בנושא תכנה

מה פירוש המונח 'תכנה'?  
תכנה היא השם הכללי הניתן לתכניות המחנות למחשבים כדי שיפעלו. חמרה היא המונח למכונות הממשיות – המחשבים עצמם וכל התקן אחר המעורב במיחשוב.

מדוע תכנה מיוצרת על סרטי קלטות?  
סרטי קלטות נוחים לשימוש ואינם דורשים ציוד מיוחד. כל מה שאתה זקוק לו לצורך טעינת סוג תכנה זה הוא רשמקול קלטות רגיל.

כיצד נשמעות תכניות מוקלטות על סרט?  
השמע תכנית מוקלטת ברשמקול ללא חיבור לספקטרום. אתה תשמע צפצוף חד. הצפצוף נגרם עקב אותות הקוד אשר מופנים לרשמקול הנמצא ברשמקול במקום למחשב. האותות נשלחים מהקלטת למחשב במהירות כה גבוהה עד כי לא ניתן להבחין בין הקולות הבודדים.

האם קיימים סוגי תכנה אחרים?  
כן. ניתן להשיג תכניות מוקלטות על מחסניות רום במקום קלטות. המחסנית מתחברת לממשק אשר מתאים למחבר בחלק האחורי של הספקטרום. תכנית במחסנית רום נטענת מיידית ללא כל המתנה.

כמו כן ניתן להשיג תכנה על תקליטון כוון זעיר. התקליטון מכיל תכניות שהוקלטו בצורה מגנטית כמו על סרט קלטות. כמה תכניות יכולות להמציא יחד על התקליטון, ולא כמו בסרט קלטות, כל תכנית ניתנת לטעינה תוך שניות במקום דקות. תקליטונים מופעלים בעזרת יחידת כוון זעיר (ראה עמ' 46).

מהו הרשמקול הטוב ביותר?  
הספקטרום יפעל היטב עם כל רשמקול נייד חזק, רצוי כזה שמחובר לרשת החשמל ולא לסוללות. למכשיר צריכה להיות בקרת עצמת קול עצמאית אך אין הכרח בבקרת טון. קיימים גם רשמקולים מיוחדים למחשבים, אלה מתוכננים לאגירה וטעינת תכניות בצורה אמינה יותר ממכשירים רגילים. נגן קלטות (Cassette deck) המהווה חלק ממערכת שמע גדולה מסובך יותר לחיבור למחשב. בנוסף לכך, שקעי תפוקת הקול בגנן קלטות אינם מפיקים אותות חזקים מספיק לספקטרום.

האם תכניות מוקלטות דורשות טיפול מיוחד?  
כמו כל צורה אחרת של איחסון מגנטי, תכניות על קלטות עלולות להזקק משדות מגנטיים חזקים, לכן אל תאחסן את הקלטות ליד מכשיר הצורך זרם חשמלי חזק. קלטות תכנה חייבות גם להשמר במקום נקי מאבק.

האם כל סוג של תכנה יפעל?  
לא. רק תכנה המיוצרת עבור הספקטרום ZX או ספקטרום ZX+ ניתנת לטעינה.



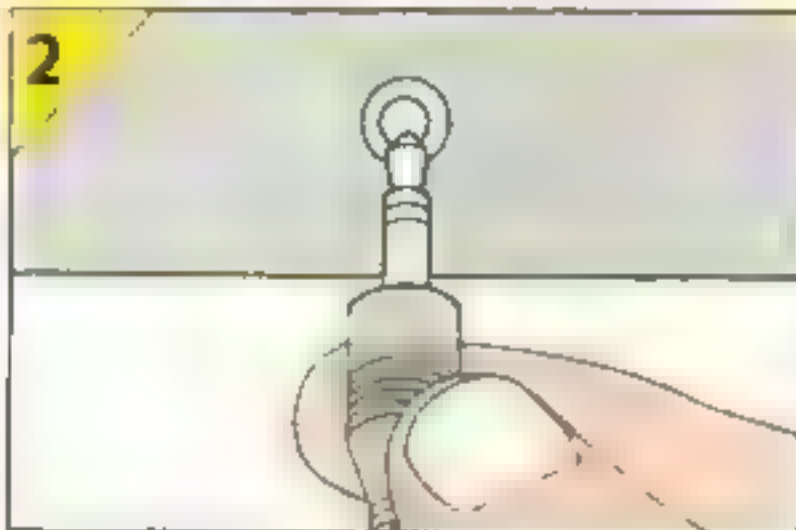
או מכובים בזמן החיבור, למרות שכדאי להוציא את הקלטות מהרשמקול לפני הפעלתו או כיבוי. כך תשמור על התכניות המאוחסנות בקלטות.

כבל הרשמקול המסופק עם הספקטרום מיועד לחיבור רשמקול קלטות. זהו כבל עם זוג תקעים קטנים בכל קצה. הצב את הרמקול ליד הספקטרום וחבר את הכבל כמתואר בציורים. הרשמקול והספקטרום יכולים להיות מופעלים

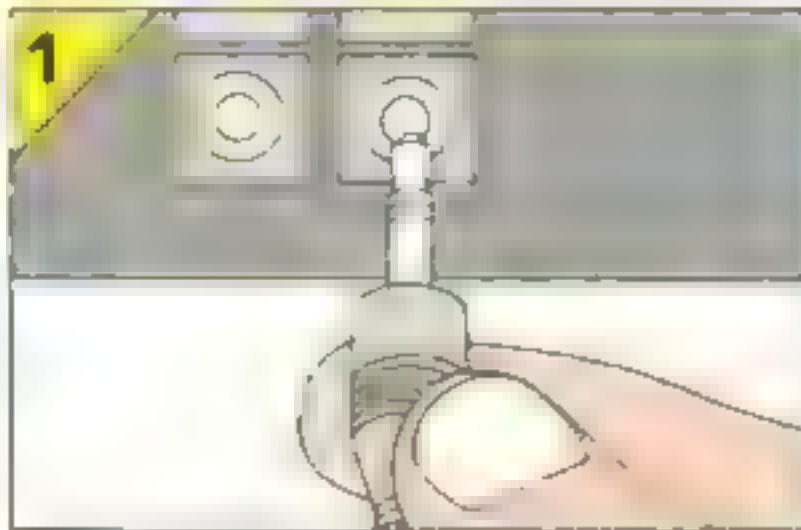
## כיצד לחבר נכון

## עצות תכנה

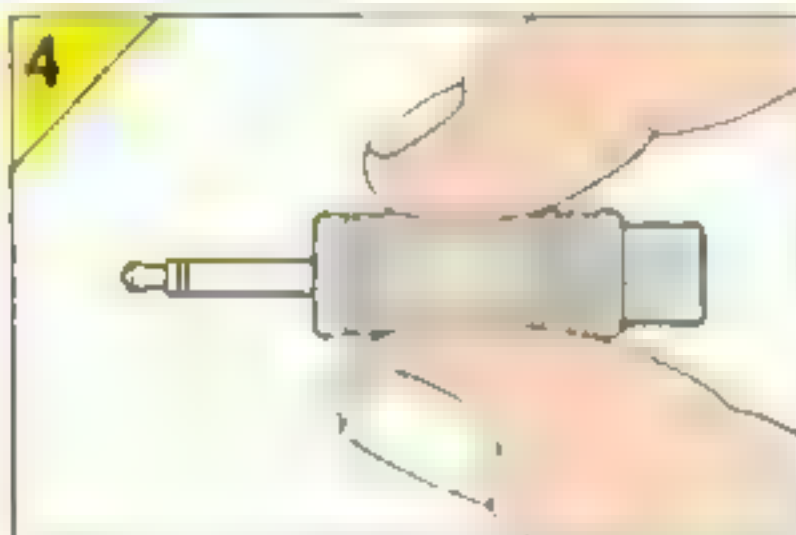
■ כבל הקלטות בספקטרום כולל תקעים צבעוניים כדי למנוע חיבורים מוצלבים מוטעים בין שקעי המחשב והרשמקול. כאשר אתה משתמש ברשמקול עם הספקטרום, השתדל תמיד להצמד לאותה מערכת, עם צבע אחד לשקעי EAR והאחר לשקעי MIC. ■ סוגי רשמקולים מסוימים עלולים להיות מושפעים מצידוד חשמלי סמוך. לפעמים הרבר עלול לגרום לעיוות האותות הנשלחים בין המחשב לרשמקול וכתוצאה מכך התכניות לא ייטענו היטב. אם הרשמקול לא פועל היטב לעיתים קרובות למדי, נסה להעבירו כך שלא יהיה ליד מקלט הטלביזיה או המחשב.



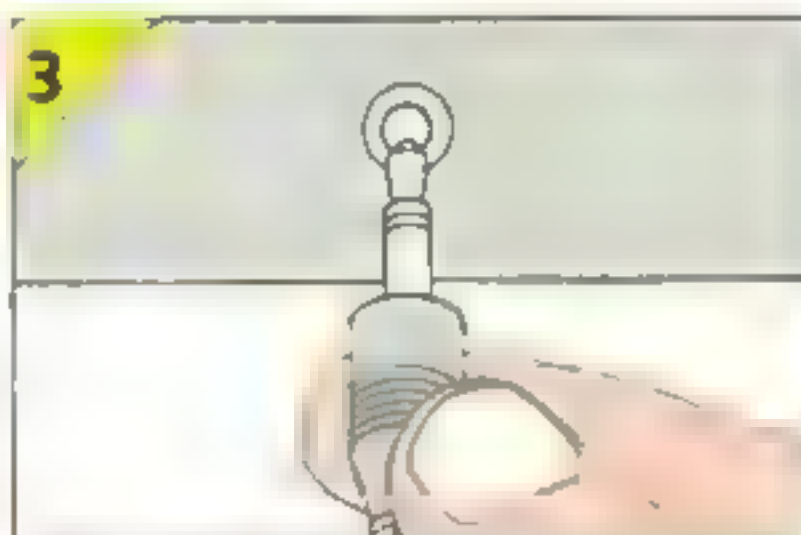
הכנס את התקע השני באותו צבע לשקע EAR ברשמקול, אם הוא קיים.



הכנס אחד מארבעת התקנים לשקע EAR בחלק האחורי של הספקטרום.

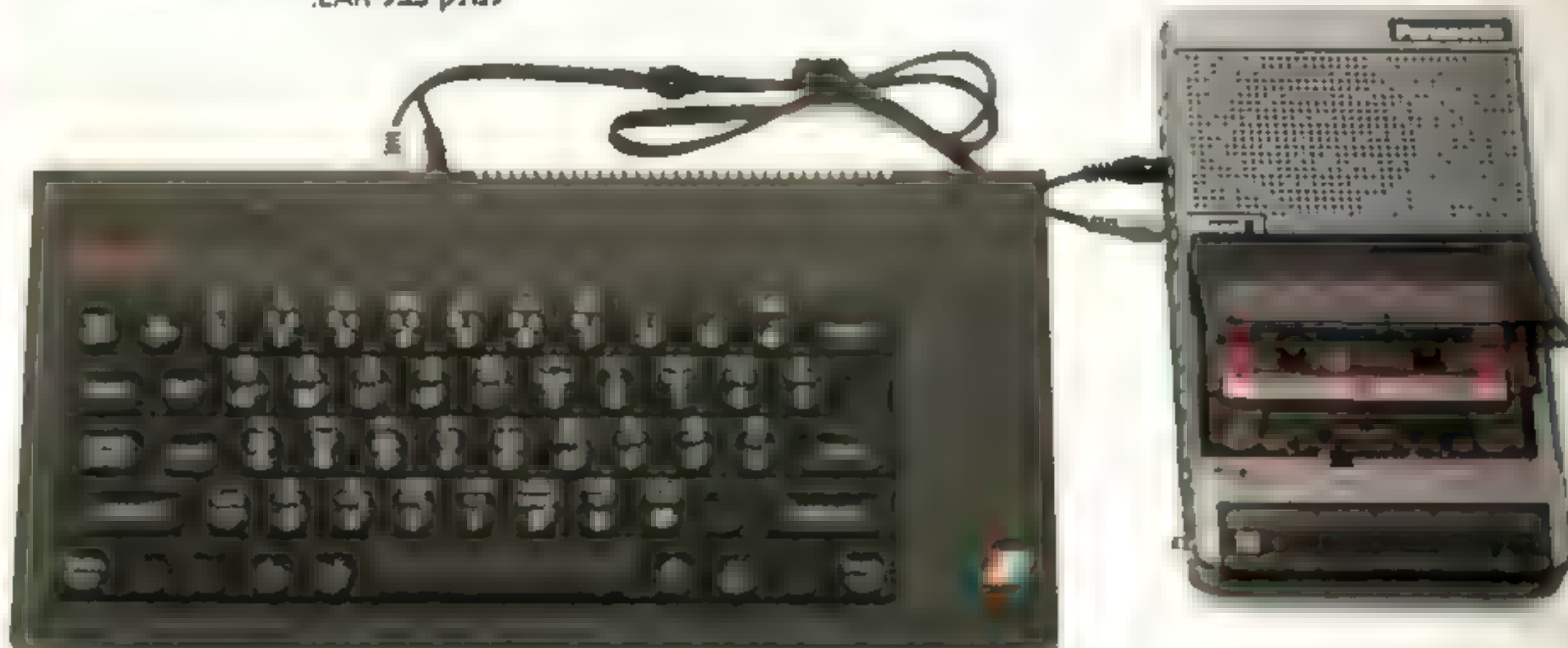


אם התקע בכבל הקלטות אינו מתאים לשקע ברשמקול, חזקק למתאם או כבל מיוחד עם התקעים המתאימים שתוכל להשיג בחנות לדברי אלקטרוניקה. שקע EAR בספקטרום דורש תקע פין 3.5 מ"מ ואות קלט של כ"ו וולט.



אם אין שקע EAR ברשמקול, חבר את התקע לשקע אזניות אם הוא קיים. במקרה אחר, נסה לחברו לשקע רמקול חיצוני.

שקעי EAR MIC כאשר טוענים תכניות אתה יכול לחבר את שקעי EAR MIC, כמתואר כאן. אך כאשר אתה מקליט תכניות (ראה עמ' 38) חובה לנתק כבל EAR.





# כיצד לטעון תכנית

כעת, לאחר שחיברת את הרשמקול לספקטרום, אתה מוכן לטעון ולהריץ תכנית. אתה יכול להשתמש בסרט תכנה מוכנה לריצה או בסרט המכיל את התכניות שכתבת. הנוהל זהה בשני המקרים. הפעל את רשמקול הקלטות. ודא שהספקטרום מופעל, והכנס את הקלטת הרשמקול. אם קיימת כבר תכנית במחשב, המתן עד שתסתיים או שתוכל לעצרה בלחיצת מקש BREAK. כעת תוכל להקיש

NEW או ללחוץ על כפתור RESET כדי למחוק את התכנית מזכרון הספקטרום, אך אין זה הכרחי מכיוון שטעינת תכנית חדשה מנקה תחילה את הזכרון. חשוב לזכור שאם אתה טוען תכנית, התכנית הקודמת תמחק מהזכרון. כעת עקוב אחר ההוראות הממוספרות. אם משהו אינו כשורה, עיין במאתר תקלות טעינת תכנה, בעמ' 16.

1 הכנס את הקלטת וגלגל אותה לנקודת ההתחלה

1

2 הצב את פקדי עצמת קול ובקרת טון של הרשמקול ברמות הדרושות. נסה להציב עצמת קול בערך של 2/3 מעוצמה מירבית ואם יש בקרת טון, הצב אותה ב-ireble מירבי.

2

3 לחץ על J ועל המסך יופיע LOAD כעת הקש את שם התכנית בין מרכאות כפולות. לדוגמא, LOAD "PROG 1"

3



4 לחץ ENTER, המסך יתרוקן.

4

5 הפעל את הסרט. גבול המסך יקבל צבע אדום או כחול או הבזקי אדום וכחול. תופעה זו מציינת שהספקטרום מחפש תכנית.

5



6 לאחר מספר שניות, יתחילו לנוע מעלה ומטה באיזור גבול המסך פסים אדומים וכחולים. זה מציין שהספקטרום התחיל לקלוט אותות.

6

## עצות לטעינת תכנה

להלן מספר עצות שיעזרו לך לחסוך זמן בעת הטעינה

1 כתוב תוויות זיהוי לכל הקלטות כך שתוכל למצוא בקלות את התכניות. אם סרט מכיל יותר מתכנית אחת, כתוב את שמות התכניות לפי הסדר על התוויות. זכור לאיית את שם התכנית בדיוק כפי שהמחשב ידרוש

1

שם התכנית	סדר
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

2

אם לרשמקול יש מונה, העזר בו למציאת תכנית במהירות על סרט המכיל יותר מתכנית אחת בכל צד. אפס את

המונה בתחילת הסרט, ואז הקש LOAD בלוח שם תכנית כלשהי (במרכאות) שאיננה על הסרט. השמע את הסרט והספקטרום יתן שם לכל תכנית שהוא מוצא, בלי לטעון אותה. כתוב את מספר המונה על התוויות בצד שם התכנית מאוחר יותר, תוכל להגיע במהירות ובקלות לתכנית הדרושה.



7 המילה Program בלוי שם התכנית, או: Bytes בלוי שם או אות, יופיעו על המסך. זה מציין שהמחשב איתר בהצלחה את התכנית.

7

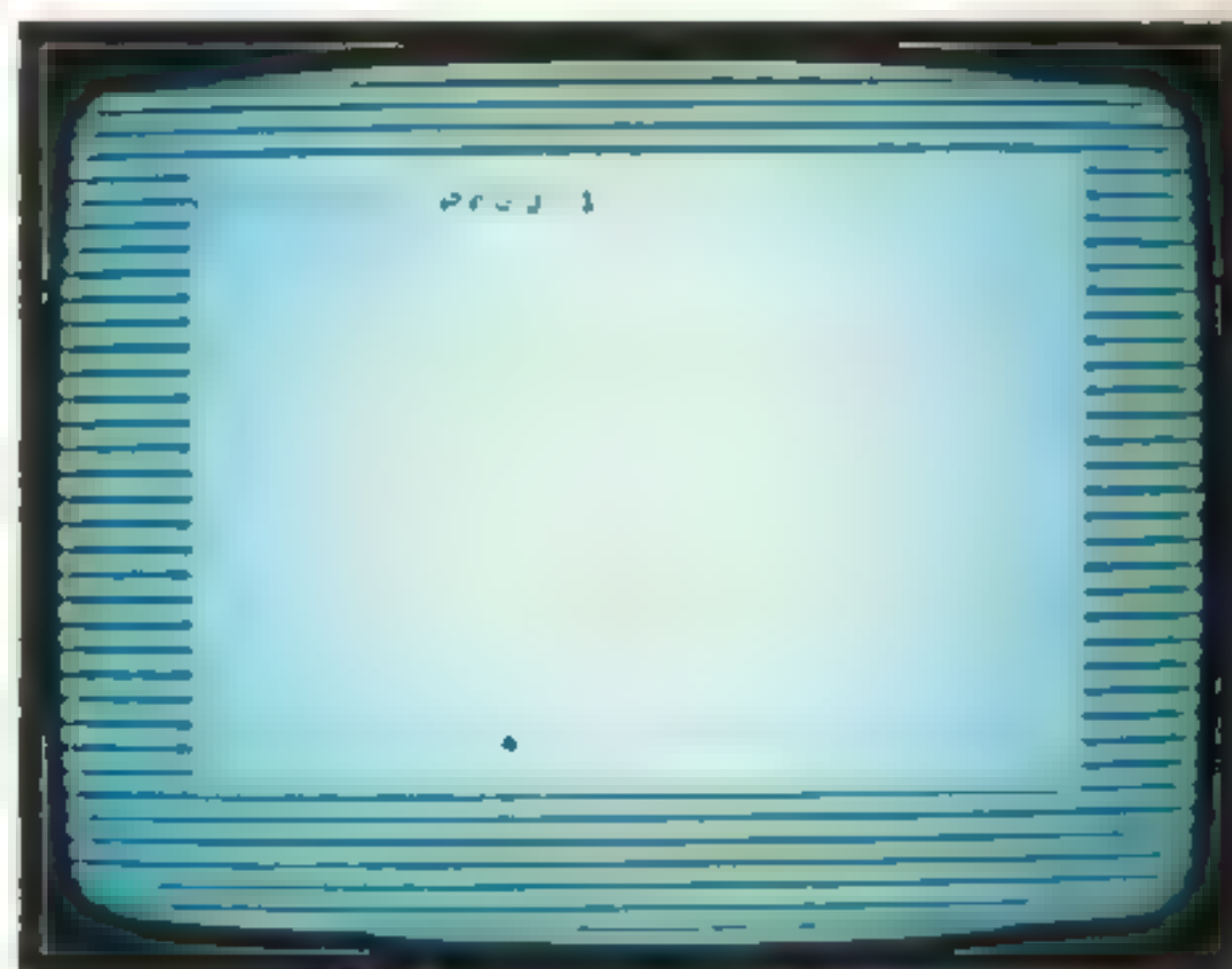


8 הפסים האדומים וכחולים מופיעים שנית כאשר המחשב ממתיך לטעינת התכנית.

8

9 בגבולות המסך מופיע רצף של פסים צהובים וכחולים. זה מציין שהספקטרום טוען את התכנית. טעינה יכולה לארוך מספר דקות אם התכנית ארוכה מאד.

9



10 פעולות 7, 8 ו-9 עשויות לחזור מספר פעמים אם התכנית מפוצלת לכמה חלקים.

10

11 התכנית עשויה להתחיל באופן אוטומטי עם גמר הטעינה. זכור לעצור את הסרט.

11

12 אם התכנית לא מתחילה לרוץ באופן אוטומטי כאשר נטענה, המסך יתרוקן ויופיע הרווח 0 OK, 0 1. עצור את הסרט.

12

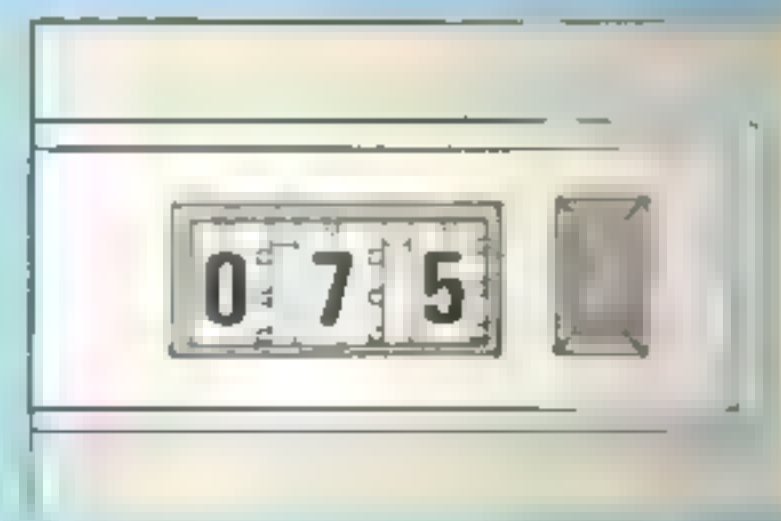


13 לחץ (RUN) R או ENTER, והתכנית תתחיל בעת לרוץ.

13



התכנית במרכזות. אסור שיהיה רווח בין שני סימני המרכזות. הספקטרום יטען את התכנית הראשונה שימצא בסרט. אם שם התכנית המופיע אינו זה שרצית, לחץ BREAK, גלגל הלאה ונסה שנית.



3 אם הסרט נמצא בתכנית הנכונה או אם אינך יודע את שם התכנית, הקש "LOAD" במקום LOAD המלווה בשם

3

VOLUME

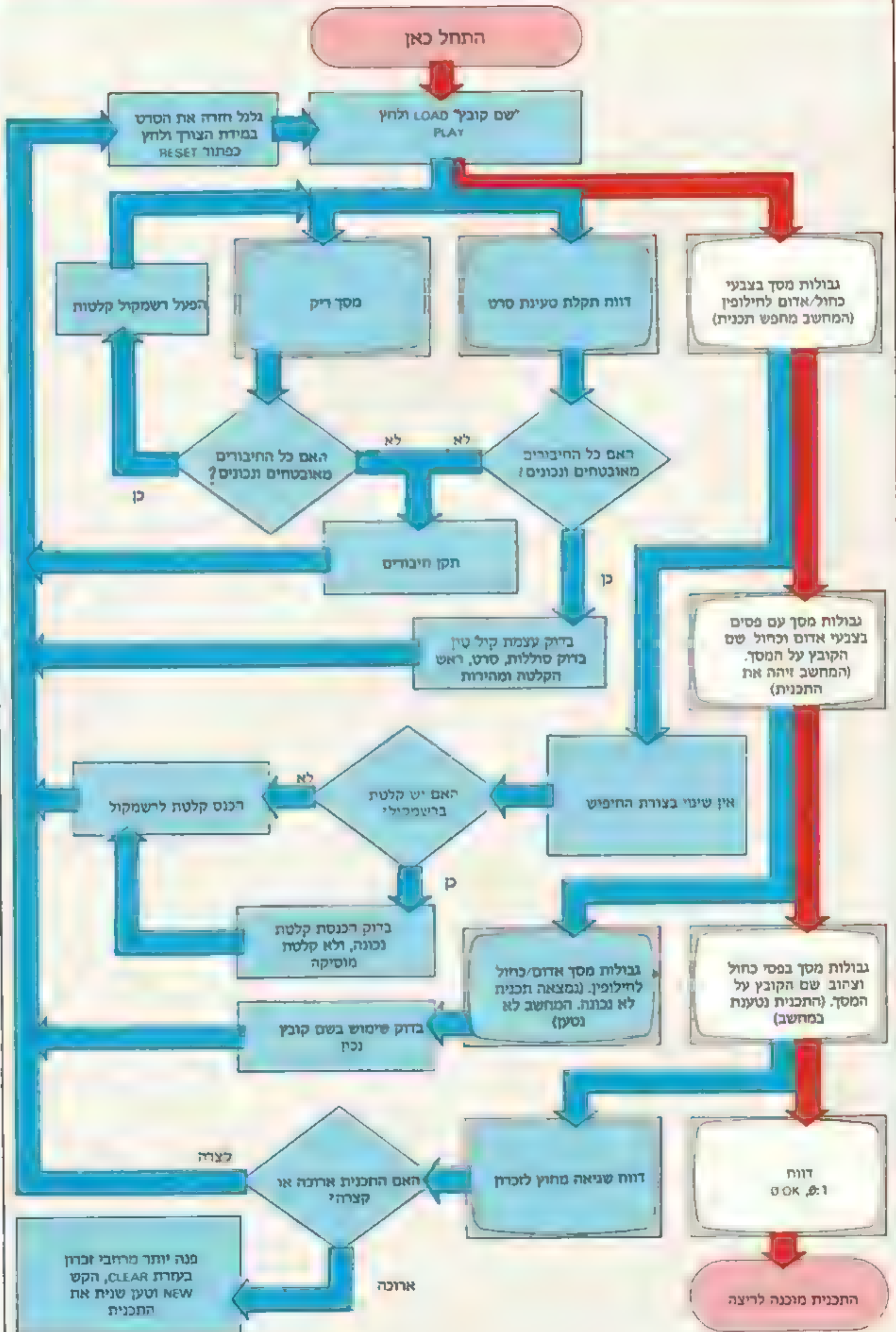
TONE



4 שים לב לרמות עצמת הקול ובקרת טון המאפשרות לספקטרום טעינת תכניות. הצב רמות אלה ברשמקול לפני הטעינה.

4







# צעדים ראשונים בתכנות

פרק זה מהווה הקדמה לכתיבת תכניות עבור הספקטרום ZX+. הפרק מסביר כיצד תוכל להכיר את הספקטרום דרך המקלדת. אח"כ תראה כיצד תתחיל להפעיל את הספקטרום לצרכיך. התכניות הקצרות המובאות כאן עבורך מתרכזות בתכונות המיוחדות של הספקטרום, כך שכאשר תתחיל בכתיבת תכניות משלך תוכל לנצל את מלוא יכולתו של המחשב.





# המקלדת – לוח הבקרה של המחשב

הניב (הגירסה) של בייסיק אותו מבין הספקטרום הוא צורה פשוטה של שפה זו אך דבת עצמה. היא מתוכננת להיות דומה עד כמה שיותר לשפה האנגלית כדי להקל על העבודה. בנוסף לכך, לספקטרום יש תכונה אחת גדולה המקלה מאוד על התכנות. זוהי שיטת הכנסת מילות מפתח בהקשת מקש אחת.

## מקשים ומילות מפתח

מילות מפתח הן מילים מיוחדות בשפת בייסיק המורות למחשב לבצע משהו – מילים כמו PRINT ר INPUT. במרבית המחשבים עליך להקיש כל אות של מילת מפתח כפי שנעשה במכונת כתיבה, ועליך לאיית במדויק כל מילה. לעומת זאת בספקטרום אתה לוחץ על מקש אחד בלבד לקבלת מילת מפתח מלאה על המסך.

מקשי תצוגת צבע – ששה מקשים אלה מייצרים מילות מפתח המבקרות את דרך הצגת הצבע על המסך.

מקשי מספרים – בנוסף ליצירת מספרים, מקשים אלה יכולים לייצר קודי בקרה בתכניות עבור הצבעים המתוארים – ראה עמ' 33. מילות המפתח הנמצאות מיד מעל ראשי המקשים מ'4 עד 8, פרט למקש 8, משמשים לבקרת הכוון הזעיר ZX בלבד.

BREAK – מקש זה עוצר ריצת תכנית. הוא לא מוחק את התכנית מהזכרון.

ENTER – לחץ על מקש זה כדי להכניס שורת תכנית לזכרון הספקטרום. משמש גם להזנת מידע למחשב בזמן תכנית.

SYMBOL SHIFT – החזק מקש זה לחוץ והקש אות או מספר לבחירת מילת המפתח או הסימן התחתונים בחלק המורם של כל מקש. כאשר הוא נלחץ אחרי EXTEND MODE הוא בוחר את הסימן או מילת מפתח הנמצאים מיד מעל החלק המורם של המקש – ראה עמ' 20-21.

בקרי סמן – לחיצה על מקשים אלה גורמת לסמן לנוע בכוח החיצים. מקשים אלה נמצאים לרוב בשימוש תכניות לבקרת תנועת צורות על המסך. כמו כן משתמשים בהם לצורך עריכת תכניות.

מקש רווחים – מקש זה מייצר רווח בדומה למקש הרווחים במכונת כתיבה.





## בחירת מילות מפתח וסימנים

במקלדת הבייסיק יש שני מקשים בעלי שימוש רב ביותר. אלה הם מקשי EXTEND MODE ו-SYMBOL SHIFT, אשר מאפשרים לך לבחור מי מתוך כל מילות המפתח והסימנים שעל מקשים אחרים יופיעו על המסך. היתה לך כבר התנסות קצרה בהקשה על מקשים אלה בעמ' 8. כעת, לאחר שתרגלת את עצמך בהכרת המקלדת, שני העמודים הבאים יראו לך כיצד לבחור כל מה שמופיע על המקלדת. לאחר שתכיר את כל המקשים, תוכל להתחיל לכתוב תכניות בעצמך.

**GRAPH** – מקש זה משמש לבחירת הצורות או תווים גרפיים על מקשים 1 עד 8. אם אתה לוחץ על מקש זה ואח"כ לוחץ על מקש מספר עם או בלי מקש CAPS SHIFT, יופיע תו גרפי על המסך.

**NEW** – מקש זה מנקח את מרחב הזכרון של בייסיק במחשב, כשהוא מוחק כל תכנית הנמצאת שם.

**DELETE** – במקש זה תשתמש אם לחצת על מקש לא נכון וברצונך למחוק מילת מפתח, אות, מספר או סימן – ראה עמ' 10.

**EDIT** – מקש זה משמש לשינוי שורה בתכנית ללא צורך לכתבה שנית בשלמותה – ראה עמ' 21.

**EXTEND MODE** – מקש זה בוחר את מילת המפתח העליונה מעל החלק המורם בכל מקש. כאשר הוא מלווה ב-SYMBOL SHIFT ומקש, הוא בוחר את הסימן או מילת המפתח הנמצאים מייד מעל החלק המורם של המקש – ראה עמ' 20-21.

בייסיק של סינקלייר יש למעלה מ-80 מילות מפתח, הנכתבות בעזרת 36 מקשים בסך הכל (26 מקשי אותיות ו-10 מקשי מספרים). מכיוון שהספקטרום משתמש במבחר גדול כל כך של פקודות בייסיק, מקשים רבים מייצרים לא אחת כי אם כמה מילות מפתח שהמחשב מכיר. מרבית המקשים נותנים למעשה מילות מפתח יחד עם אות, מספר, סימן או אפילו צורה (תו גרפי), אשר ניתן להשתמש בהם בתכניות.

**TRUE VIDEO** ו-**INV VIDEO** רק מקשים אלה מכניסים קודי בקרה לשורות תכנית כדי ליצור צבעים רגילים או הפוכים.



**CAPS SHIFT** – לחץ על מקש זה עם מקש אות ליצירת אות גדולה. אם אתה רוצה מספר אותיות גדולות, השתמש ב-CAPS LOCK.

**CAPS LOCK** – השתמש במקש זה אם תרצה שמקשי האותיות יצרו אותיות גדולות כל הזמן. לחץ שנית לקבלת אותיות קטנות.

**BEEP** מקש זה מייצר את מילת המפתח המבקרת את הסינטיסייזר הקולי של הספקטרום.



# כיצד להפעיל את המקשים

אתה יכול לקבל עד 6 מילות מפתח, אותיות, מספרים או סימנים שונים ממרבית המקשים בספקטרום ZX+. ברם, בחירת אות או מילה מפתח על המקלדת אינה כה מסובכת לאחר שהכרת את אחת מתכונותיו המיוחדות של הספקטרום. אם אתה לוחץ על מקש, התוצאה שתופיע על המסך תלויה באופן הפעולה בו נמצא המחשב באותו זמן. כל אופן פעולה מאפשר הקשת סוגי מידע אחרים, כמו מילות מפתח, אותיות או תווים גרפיים. היתרון בכך הוא, שבזמן הפעלת המקלדת, הספקטרום בעצם עוזר לך בבחירת אופני פעולת המקלדת כך שתקיש פקודות ונתונים בסדר הנכון. בשני עמודים אלה תמצא מה בדיוק עושים כל אופני הפעולה.

## אופן פעולה – מילת מפתח

הפעל או אתחל את הספקטרום כך שתופיע הודעת זכויות יוצרים. כעת לחץ ENTER. בפניה השמאלית תחתונה יופיע K מהבהב. הריבוע המהבהב נקרא סמן. הוא מראה לך היכן עתיד להופיע משהו על המסך והאות K מציינת שהמחשב נמצא באופן פעולה של מילת מפתח (keyword). לחץ על מקש אות כלשהו ומילת המפתח העליונה בחלק המורם של המקש תופיע על המסך. נסה Q, למשל, ומילת המפתח PLOT תופיע. לחץ על מקש DELETE למחיקת מילת המפתח ונסה מקשים אחרים. מקשי מספרים יתנו מספרים, אך אם תלחץ על מקש אות, תופיע מילת המפתח העליונה בחלק המורם של המקש.

השתמש שנית ב-DELETE כך שהסמן K יופיע שוב. כעת לחץ על אחד משני מקשי SYMBOL SHIFT, החזק אותו לחוץ ולחץ על מקש אות כלשהו. הפעם, יופיעו על המסך מילת המפתח או הסימן הנמצאים מעל האות בחלק המורם של המקש. במקש מספר, יופיע הסימן בצד ימין בחלקו המורם של המקש. כלומר, אופן הפעולה – מילת מפתח מתייחס לחלק המורם של המקש.

## כיצד לבחור מילת מפתח, סימן או תו

תצטרך לשנות לאופן הפעולה הנכון. התבונן תחילה בסמן על המסך כדי לזהות את אופן פעולת המחשב.

מקש, שים לב למקומה על המקש, ובעזרת שתי הדוגמאות שלהלן קבע אילו מקשים – אם קיימים –

כאן תוכל לראות כיצד לבחור כל מילה מפתח, סימן או תו על מקש או מקש מספר. בזמן בחירת פונקציית

מקש מספר	
אופן פעולה (א) מילת מפתח	
מקש בלבד	BORDER
ומקש	SYMBOL SHIFT
אופן פעולה (E) מורחב	
ומקש בלבד	BIN
ומקש	BRIGHT
ומקש	EXTEND MODE
אופן פעולה (L) אותיות	
מקש בלבד	b
ומקש	B
ומקש	SYMBOL SHIFT
אופן פעולה (C) אותיות גדולות	
ומקש בלבד	B
ומקש	SYMBOL SHIFT
אופן פעולה (G) גרפיקה	
עם מקשים A עד U בלבד עם גרפיקה משתמש.	GRAPH

מקש מספר	
אופן פעולה (א) מילת מפתח	
מקש בלבד	3
ומקש	#
אופן פעולה (E) מורחב	
ומקש בלבד	magenta
ומקש	LINE
אופן פעולה (L) אותיות	
מקש בלבד	3
ומקש	#
אופן פעולה (C) אותיות גדולות	
ומקש בלבד	3
ומקש	#
אופן פעולה (G) גרפיקה	
ומקש בלבד	GRAPH
ומקש	GRAPH



תחילה, והחזק אותו לחוץ, יחד עם לחיצת מקש האות על מקש B, לדוגמה, תקבל עתה BRIGHT כלומר, במצב פעולה – מורחב תקבל את זוג מילות המפתח מעל החלק המורם של המקש. לאחר לחיצת מקש (או EXTEND MODE) באופן פעולה – מורחב, המחשב חזר אוטומטית לאופן פעולה של אותיות או אותיות גדולות.

#### אופן פעולה – גרפיקה

אופן הפעולה החמישי נקרא גרפיקה, ונוצר בלחיצת מקש GRAPH. הסמן משתנה ל-G. לחץ על מקשים 1 עד 8 וראה כיצד התווים הגרפיים המסומנים על מקשים אלה יופיעו על המסך. כעת לחץ CAPS SHIFT ומספר כלשהו בין 1 ל-8. הגרפיקה מופיעה שנית, אך הפעם שחזר ולבן מתהפכים. כדי לצאת מאופן הפעולה – גרפיקה, עליך ללחוץ שנית על GRAPH, מכיוון שהמחשב לא יוצא ממנו בצורה אוטומטית.

#### אופני הפעולה – אותיות ואותיות גדולות

לאחר יצירת מילת מפתח או סימן באופן הפעולה – מילת מפתח, המחשב משנה אוטומטית את הסמן ל-L. הסמן נמצא כעת באופן פעולה – אותיות לחץ על מקש אות כלשהו והאות הקטנה תופיע. לחץ על מקש מספר והמספר יופיע. אם ברצונך להקיש אות גדולה, החזק מקש CAPS SHIFT לחוץ ולחץ על מקש האות.

אם ברצונך להקיש רק אותיות גדולות, לחץ CAPS LOCK תחילה. הסמן עובר למצב C. הספקטרום נמצא עתה באופן פעולה – אות גדולה ואתה תקבל רק אותיות גדולות על המסך. במצב אותיות גדולות ניתן עדיין לקבל מספרים. כדי לחזור לאופן פעולה – אותיות (L), לחץ שנית על CAPS LOCK.

#### אופן פעולה – מורחב (E)

אופן הפעולה – מורחב נוצר ע"י לחיצת מקש EXTEND MODE. הסמן משתנה עתה ל-E. לחץ על מקש אות כלשהו, ועל המסך תופיע מילת המפתח העליונה בזוג מילות המפתח שמעל החלק המורם במקש. לדוגמה, לחץ B ותקבל BIN. לקבלת מילת המפתח או הסימן התחתונים מעל חלק המקש המורם, לחץ על אחד ממקשי SYMBOL SHIFT.

#### עריכה בספקטרום

כאשר אתה נותן פקודות או מגיע לשלב כתיבת התכניות לספקטרום, תרצה בודאי לתקן טעויות בפקודות או שורות תכנית או לשנותן. ניתן לעשות זאת בקלות בעזרת תכנית העריכה.

##### כיצד לתקן טעות

אם אתה מנסה להכניס שורה או פקודה שגויה בביסיק, הספקטרום יציג סימן שאלה (?) מהבהב לפני השגיאה. כדי לתקן את הטעות לחץ על מקשי בקרת הסמן לכוח ימין או שמאל כדי להביא את הסמן לצד ימין של הטעות. כעת ניתן למחוק את הטעות ע"י הקשת DELETE או הוספת האות, מילת מפתח, מספר או סימן הנדרשים. אח"כ לחץ ENTER. לדוגמה, נניח שאתה רוצה שהמחשב יבצע 7 כפול 8 ולא לחצת על SYMBOL SHIFT בעת הקשת סימן \* מה שהקשת בעצם הוא

PRINT 7\*8

הספקטרום לא יכול לבצע פקודה זאת, ולכן בלחיצת ENTER יופיע סימן שאלה מהבהב לפני האות B, שם נמצאת השגיאה. כל שעליך לעשות הוא להביא את הסמן לימין הטעות וללחוץ DELETE כדי למחוק את ה-B.

אח"כ לחץ SYMBOL SHIFT ומקש B לקבלת הסימן \*, בלוי ENTER כדי שהמחשב יציית לפקודה הנכונה. אינך חייב להזיז את הסמן לסוף השורה תחילה, הספקטרום מבצע את הפקודה ומציג את התוצאה.

##### כיצד לערוך שורת תכנית

כאשר אתה כותב תכנית, אתה בונה רצף של שורות פקודות ממוספרות הנקרא רישום. אם לאחר כתיבת התכנית, אתה 'דוש' זאת ע"י לחיצת (LIST) ENTER, תראה סימן > כנגד אחת משורות התכנית. אם לא, לחץ על מקש בקרת הסמן בכוח מעלה או מטה. אם תלחץ כעת על EDIT, שורה זאת תשוכפל בתחתית המסך וניתן יהיה לשנותה כמקדם בעזרת מקשי הסמן DELETE. לחץ ENTER כדי להכניס את השורה החדשה לתכנית. אם ברצונך לערוך שורה אחרת, הזז את סימן > בעזרת מקש בקרת סמן מעלה או מטה עד לשורה הטעונה שינוי והקש EDIT. אם התהליך ארוך מדי, הקש LIST מלווה במספר השורה ואז הקש EDIT. בכל מקרה, השורה אותה זרדת תופיע בתחתית המסך וניתן יהיה לשנותה. כדי למחוק שורה שלמה מתכנית, פשוט הקש פנימה את מספר השורה בלבד ואח"כ לחץ ENTER. אם אתה מריץ תכנית המכילה שגיאה, תראה דו"ח מסך. דוחות אלה מוסברים בעמ' 74.





# מחשבון הטלביזיה

## התכנית הראשונה שלך

לאחר שפקודה התבצעה, הספקטרום שוכח אותה. אם אתה רוצה שהמחשב יחזור על החישוב, ניתן לכתוב אותו כתכנית. הקש פקודה זו ולחץ ENTER:

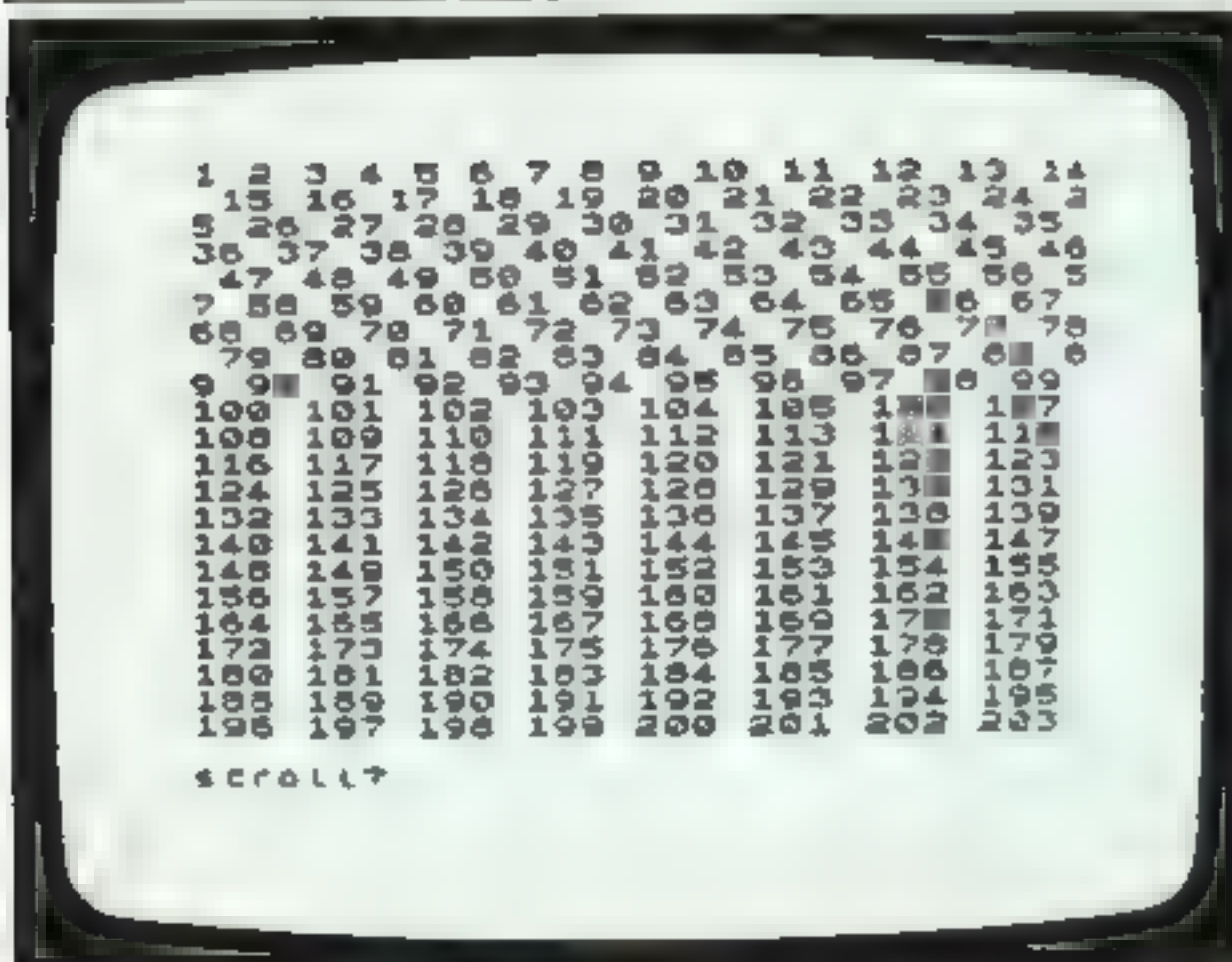
10 PRINT 6 + 2

הפעם, פקודה זו לא מתבצעת מייד. המחשב מציג במקום זאת את הפקודה על המסך. כעת הקש R(RUN) R(ENTER). התוצאה 8 תופיע עכשיו.

כל הפקודה מהווה עתה תכנית מחשב. הצבת מספר בראש השורה גורמת לספקטרום למקם את הפקודה בזכרון, אך הוא אינו מבצע אותה אלא לאחר שהתבקש לעשות זאת. כאשר אתה מריץ תכנית על הקשת R(RUN) R(ENTER), הפקודה מתבצעת. ההוראה למחשב נקראת עתה הצהרה במקום פקודה, והיא מהווה שורה ממוספרת בתכנית. הצהרות תכנית מתבצעות תמיד לפי סדר מספרי השורות, ואלה עולות בדרך כלל בעשרות כך שניתן יהיה בעתיד להכניס שורות נוספות ביניהן במידת הצורך. כעת, נגרום לספקטרום לעבוד קשה יותר. הקש את התכנית שלהלן. זכור ללחץ ENTER לאחר הקשת כל שורה, וכאשר סיימת, לחץ R(RUN) R(ENTER). לאחר שהדצת את התכנית, זה מה שאתה אמור לראות:

## טבלת מספרים

```
10 LET n=1
20 PRINT n; " ",
30 LET n=n+1
40 GO TO 20
```



כל המספרים מ-1 עד 203 מוצגים על המסך. כעת לחץ על מקש כלשהו פרט למקשי N, רווח, STOP או BREAK. מערך חדש של מספרים יופיע על המסך. תכנית זו משתמשת במשתנה. במקרה זה,

הספקטרום ZX+ יכול לבצע חישובים במהירות עצומה ובדיוק רב. כל מה שנדרש הוא כמה מספרים לעבוד עליהם וסימנים כמו + ו- האומרים לו מה לעשות עם המספרים. ראשית הקש פקודה זאת (את הסימן + תמצא על מקש K):

PRINT 6 + 2

זאת היא פקודה. כאשר אתה לוחץ ENTER, הפקודה נעלמת והתשובה, המספר 8, מופיע על המסך. הספקטרום משתמש ב-5 סימנים הידועים כפעולות אריתמטיות עבור חישובים. תוכל לראות מה מבצע כל סימן בטבלה שלהלן. אתה יכול להשתמש בכולם באותה צורה יחד עם PRINT. הקשת פקודות כמו 6 + 2 PRINT הופכת את הספקטרום למחשבון, אך הוא יכול לבצע דברים רבים שמחשבון רגיל אינו יכול לבצע. ראשית, הוא יכול להציג חישובים ואת תוצאותיהם יחד. הקש פקודה זו:

PRINT "6+2=";6+2

המחשב יגיב בהצגת:

6+2=8

מה שקורה הוא שפקודת PRINT גורמת לכל מה שנמצא בין מרכאות כפולות להיות מוצג על המסך, ולכן מופיע  $6 + 2 =$ . התווים בין סימני המרכאות מהווים מחרוזת. הסימן נקודה-פסיק מורה לספקטרום להציג את התוצאה מייד לאחר סימן השוויון.

## טבלת סימני פעולות אריתמטיות

הסימנים שלהלן או 'פעולות אריתמטיות' משמשים בספקטרום לביצוע פעולות חשבוניות. שים לב לכך שהמחשב אינו משתמש בסימנים  $\times$  או  $\div$ .

סימן	מקש פונקציה	דוגמה
+	K חבר שני מספרים	$8 + 2 = 10$
-	J חסר שני מספרים	$8 - 2 = 6$
*	B הכפל שני מספרים	$8 * 2 = 16$
/	V חלק שני מספרים	$8 / 2 = 4$
↑	K העלה מספר ראשון בחזקת המספר השני	$8 \uparrow 2 = 64$





### מדוע עליך להשתמש בסוגריים

לעיתים תצטרך להשתמש בסוגריים במהלך החישוב. הקש את שתי הפקודות הבאות והשווה את התוצאות:

```
PRINT 6+2/4
PRINT (6+2)/4
```

הראשונה נותנת 6.5 ואילו השנייה נותנת 2. הסיבה לתוצאות שונות אלה היא בכך שלמחשב יש מערכת עדיפויות פנימית לצורך חישובים. המחשב מבצע ראשית  $\uparrow$ , אח"כ  $*$  או  $/$  ולבסוף  $+$  או  $-$ , אך פעולות בסוגריים מתבצעות תמיד ראשונות. ולכן, בפקודה הראשונה שלעיל, המחשב מחלק ראשית 2 ב-4 ואחר כך מחבר את התוצאה (0.5) ל-6. בפקודה השנייה, המחשב מחבר 6 ו-2, ואח"כ מחלק את התוצאה ב-4.

### ביצוע סימני פיסוק

הספקטרום משתמש במבחר של סימני פיסוק. סימנים אלה חשובים מאחר ורבים מהם משמשים גם כהוראות למחשב, ומשפיעים על הדרך בה הוא מבין שורת תכנית או יוצר תצוגה.

- **נקודה פסיק** – כאשר משתמשים בו עם PRINT, המחשב מציג את שני הפרטים משני צידיו צמודים אחד לשני על המסך.
- **נקודתיים** – מסמן סיום הצהרה אחת בשורת תכנית ואת תחילתה של ההצהרה הבאה.
- **סימן מרכאות** – כל התווים הנמצאים בין מרכאות אינם נחשבים כמספרים או משתנים אלא כטקסט. מרכאות פותחות וסוגרות מחרוזת.
- **פסיק** – כאשר מופיע בפקודת PRINT, הוא אומר למחשב להציג את הפרט הבא באמצע השורה או בתחילת השורה הבאה. אל תשתמש בו לציון אלפים או מליונים.
- **נקודה** – נקודה עשרונית או סיום משפט.

המשתנה נקרא n. ניתן להשתמש בכל אות או מילה – n מייצג כאן פשוט מספר. למשתנה ניתן עודך אשר משתנה תוך ריצת התכנית. בשורה 10, מילת המפתח LET משמשת לקביעת עודך המשתנה כ-1. שורה 20 מציגה את הערך כשהוא מלווה ברווח. בשורה 30 משתמשים שנית בפקודת LET, והפעם להגדיל את הערך ב-1, כך ש-n הופך ל-2. שורה 40 משתמשת במילת המפתח (יחידה) GOTO כדי לשלוח את התכנית חזרה לשורה 20, המציגה כעת 2. התהליך חוזר על עצמו שוב ושוב עד שהמספרים ממלאים את המסך.

### כיצד לגרום לתכנית לבקש מספר

עצור את התכנית בלחיצת BREAK. כעת הקש פנימה שורה חדשה:

n INPUT 10

שורה זו מחליפה את שורה 10 הקודמת בתכנית. כאשר תריץ אותה, המחשב מחכה למספר שתכנים אליו. הקש פנימה מספר כלשהו ולחץ ENTER. כעת המספרים יתחילו במספר שהכנסת. הסיבה לכך היא שפקודת n INPUT קובעת את עודך n כשווה למספר שהכנסת. INPUT מורה למחשב לקלוט נתון במהלך ריצת תכנית.

### תכנות טבלת כפל

לחץ על כפתור RESET למחיקת התכנית הישנה והקש את התכנית הבאה. תכנית זו גורמת לספקטרום להכפיל מספרים. הקש מספר כלשהו ועל המסך תופיע טבלת כפל עבור מספר זה. לחץ על מקש כלשהו פרט למקשי N, BREAK או רווח, והטבלה תמשיך. לחץ BREAK והרץ את התכנית שנית ליצירת טבלה חדשה. להלן מופיעה התכנית ומה שתוכל לראות אם תקיש פנימה 3 ואח"כ 146:

### טבלת כפל

```
10 LET x=1
20 INPUT n
30 PRINT n;"*";x;"=";n*x
40 LET x=x+1
50 GO TO 30
```





# הצבע וכיצד להשתמש בו

הספקטרום ZX+ יכול ליצור 8 צבעים שונים, כאשר לכל צבע יש מספר קוד. ניתן להשתמש בכל צבע בדרכים שונות – כצבע גבול מסך, כצבע דיו, וכצבע נייר (רקע פנימי במסך).

## קודי צבע בספקטרום ZX+

טבלה זאת מתארת את הצבעים והקודים הקיימים בספקטרום. אינך חייב לזכור קודים אלה; מקשי המספרים היוצרים אותם מסומנים גם בשמות הצבעים. (שמות אלה אינם מילות מפתח).

מספר	צבע
0	טחור
1	כחול
2	אדום
3	סגול
4	ירוק
5	תכלת
6	צהוב
7	לבן

הגוונים שתקבל למעשה על מקלט הטלביזיה תלויים במקלט עצמו ובכווןן עצמת הצבע, ופקדי ניגודיות (קונטרסט) ובהירות. זכור שעליך להשתמש במקלט טלביזיה צבעוני.

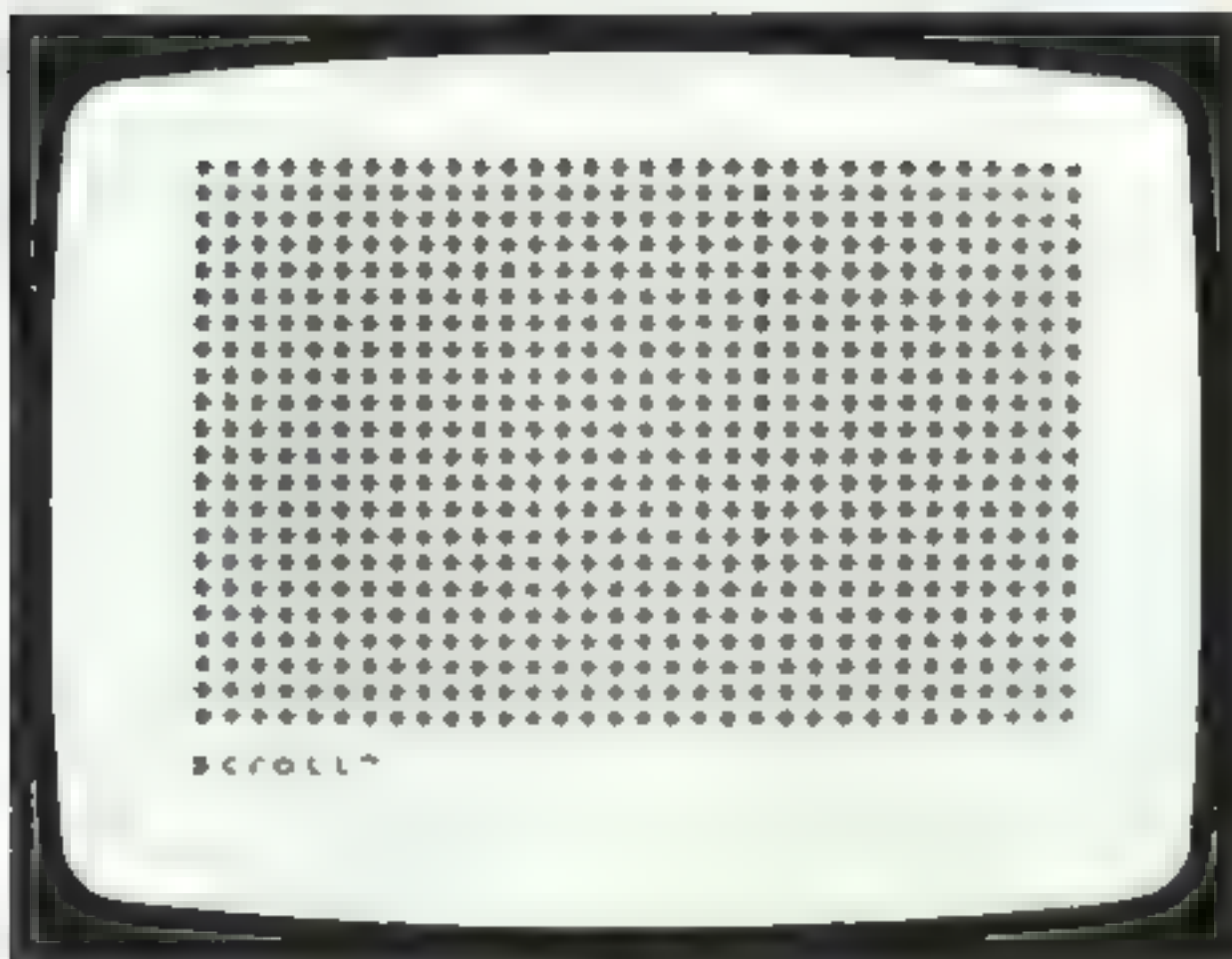
## שלושת אופני השימוש בצבע בספקטרום

ניתן לבקר את הצבע בשלוש דרכים שונות. צבע הגבול הוא צבע גבולות המסך מסביב לשטח התצוגה העיקרי. צבע הדיו הוא הצבע בו מופיעים תווים (אותיות, מספרים, סימנים וצורות גרפיים) ונקודות או קווים. צבע הנייר הוא צבע הרקע, או על פני כל שטח התצוגה או בריבוע סביב כל תו.

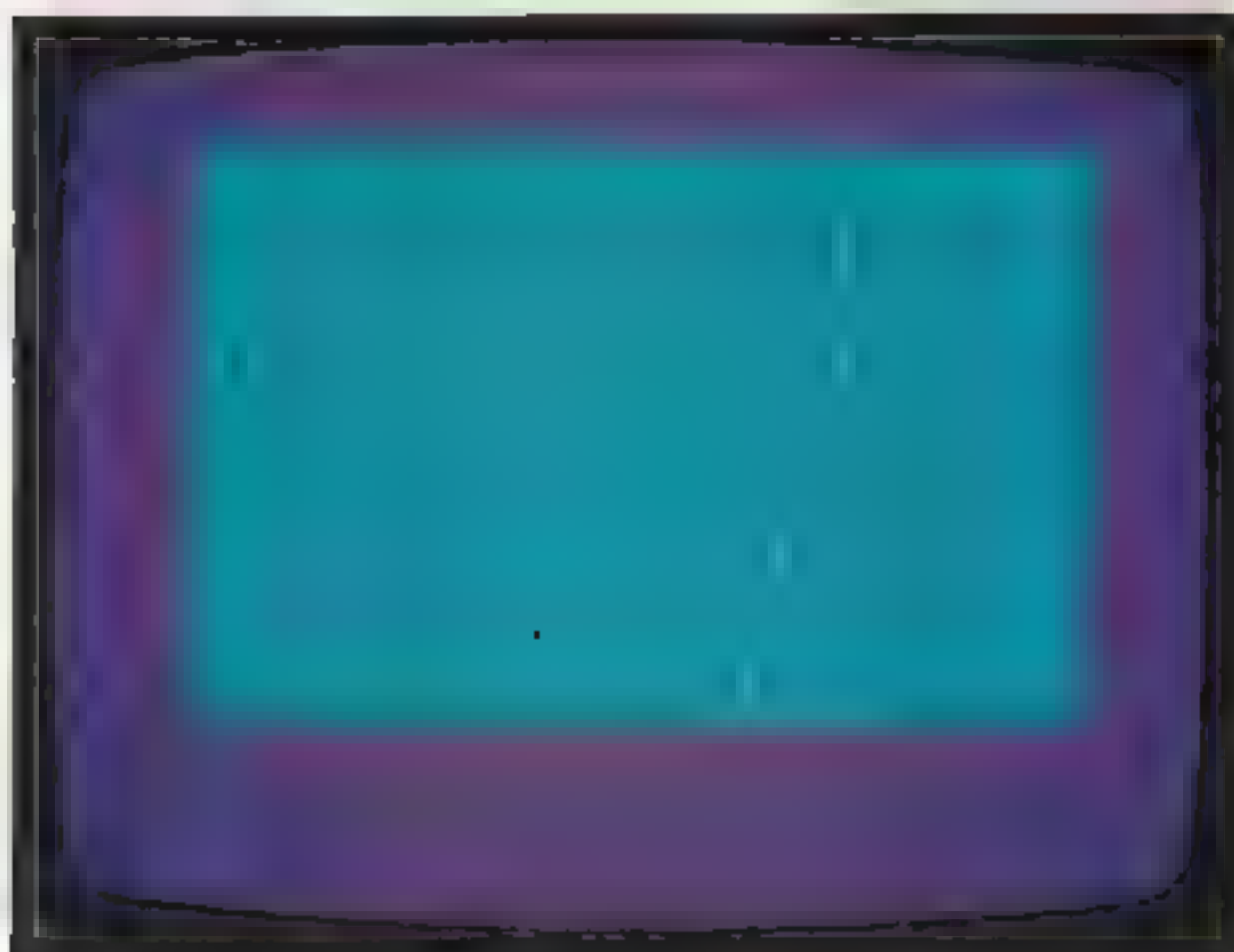
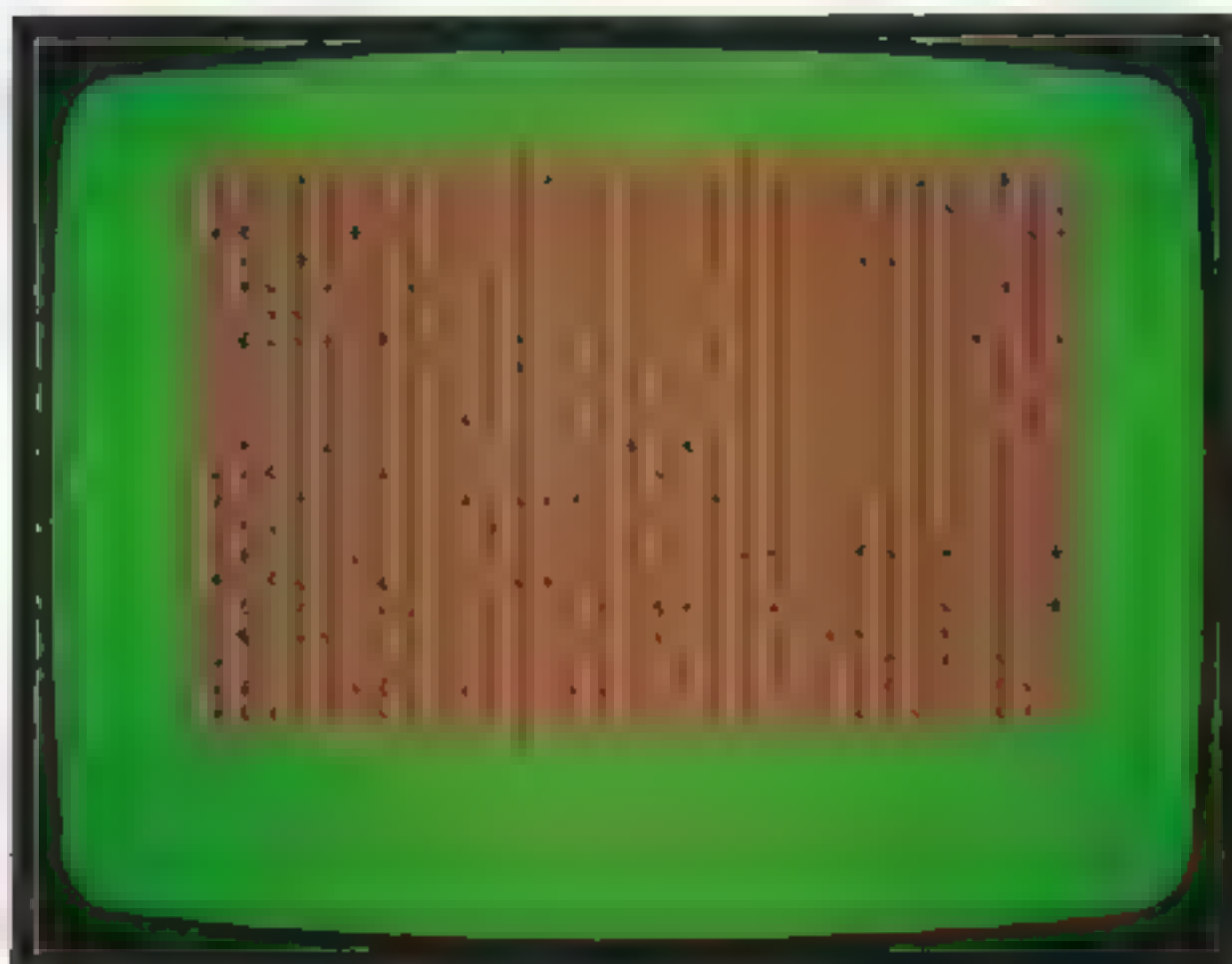
כאשר אתה מפעיל את הספקטרום, הוא משתמש בצבעים הקבועים מראש. צבע הדיו שחור, וצבעי הגבול והמסך לבנים. אתה יכול לשנות צבעים אלה כהרף עיין במתן פקודות ישירות מהמקלדת. נתקלת כבר באפשרות זאת בעמ' 6-7, בהם השתמשת בפקודת BORDER לבדיקת החיבור הנכון של הטלביזיה והספקטרום מבחינת הצבע. כעת לחץ על כפתור RESET והקש והרץ תכנית פשוטה זאת:

בדיקת צבע

```
10 PRINT "*";
20 GO TO 10
```



שורות של כוכבים נבנות בשחור ולבן. כעת לחץ BREAK והקש פנימה כמה פקודות צבע. הקש פנימה את מילות המפתח INK, BORDER, PAPER כשהן מלוות במספר בתחום 0 עד 7, לחץ ENTER אחרי כל פקודה, והרץ שנית את התכנית להלן מופיעות שתי תצוגות, הראשונה עם INK 7, PAPER 2, BORDER 4, והשנייה עם INK 1, PAPER 5, BORDER 3.





## תכנות דיאגרמות עמודות צבעוניות

התכנית הבאה משתמשת בצבעי הספקטרום ליצירת דיאגרמת עמודות. הדיאגרמה מתארת 12 טמפרטורות יומיות בעמודות צהובות עם מספרים. בשורה 60, הכנס שני רווחים בין סימני המרכאות.

### דיאגרמת עמודות

```
10 BORDER 0: PAPER 1: CLS
20 LET C=4
30 FOR X=1 TO 12
40 READ I
50 FOR L=21 TO 21-I STEP -1
60 PRINT PAPER 0; AT L,C;" "
70 NEXT L
80 PRINT INK 2; AT 20-I,C,I
90 LET C=C+2
100 NEXT X
110 DATA 20,15,13,16,19,20,18,1
1,12,19,14,17
```



כעת הוסף את השורות הבאות, והקש פנימה את שורה 110 כמתואר. הדיאגרמה תופיע עתה בשני צבעים. כדי ללמוד יותר על פקודות READ וDATA, עיין בעמ' 33.

### דיאגרמת עמודות כפולה

```
85 READ I
86 FOR L=21 TO 21-I STEP -1
87 PRINT PAPER 3; AT L,C;" "
88 NEXT L
89 PRINT INK 1; PAPER 5; AT 20-
I,C,I
110 DATA 20,6,15,4,13,5,16,6,19
,10,20,8,18,6,11,4,14,6,19,8,14,
9,17,7
```



## כיצד לכתוב תכניות עם צבע

ניתן להשתמש במילות המפתח PAPER, BORDER ו-INK בתוך תכנית כדי שטקסט, טבלאות, צורות ותמונות יופיעו בכל מיני צבעים. השימוש ב-BORDER בשורת תכנית גורם לשינוי צבע גבולות המסך ברגע שהספקטרום הגיע לשורה זו. כאשר INK מופיע בשורה משלו, הוא נותן צבע דיו חדש לכל התווים או השורות שיופיעו על המסך. פקודת PAPER בשורה משלה משנה את צבע הנייר אך רק סביב התווים (כולל גם נקודות או שורות). אם אתה רוצה לשנות את צבע כל הרקע, עליך להקיש PAPER בלוח CLS. ניתן גם להשתמש ב-INK ו-PAPER אחרי PRINT במקרה זה, רק אותם תווים המוצגים על ידי PRINT יקבלו את הצבעים מהוראות INK ו-PAPER. התכנית הבאה מדגימה את כל צבעי הגבול, דיו ונייר. כמו כן היא מראה כיצד להשתמש ב-INK ו-PAPER לאחר PRINT.

### צירופי צבע

```
10 FOR B=0 TO 7
20 BORDER B PAPER B CLS
30 PRINT AT 5,12; INK 9; B
40 FOR P=0 TO 7
50 PRINT AT P+5,5; INK P; PAPER
R 9,P
60 BEEP 0.5, B+P-20+P
70 FOR I=0 TO 7
80 PRINT INK I; PAPER P;" ";I
90 BEEP 0.01, I+5
100 NEXT I
110 NEXT P
120 NEXT B
```



כאשר תריץ תכנית זאת, תראה את כל הצירופים האפשריים של גבול, נייר ודיו. בתכנית יש 3 משתנים, B עבור מספר גבול, I עבור מספר הדיו ו-P עבור מספר נייר. BEEP יוצר את הקול, והשורות המתחילות ב-FOR ו-NEXT מציינות התחלת וסוף לולאת תכנית אשר משנה לפי הסדר את כל מספרי הצבעים מ-0 עד 7. על השימוש ב-FOR ו-NEXT בלולאות תכנית תמצא יותר בעמ' 27. שים לב לכך ש-INK ו-PAPER יכולים לקבל את הערך 9. כך הופכים את צבעי הדיו או הנייר לשחור או לבן כך שיובלטו כנגד הרקע או התו.



# גרפיקת "עשה זאת בעצמך"

הספקטרום ZX+ יכול לתת גרפיקה בהפרדה גבוהה ובהפרדה נמוכה. שני הסוגים יכולים להופיע יחד על המסך. תצוגות גרפיקה בהפרדה נמוכה בנויות מגושי צבע. בשני העמודים שלהלן, תלמד כיצד ליצור גושים אלה במקלדת, וכיצד למקם אותם על המסך.

## המסך בהפרדה נמוכה

במסך בהפרדה נמוכה יש 32 מקומות בהם ניתן למקם תוים לרוחב המסך 22 מקומות לאורך המסך. לכל נקודה במסך יש זוג מספרים המזהה אותה. ראשית בא מספר השורה, שהוא מספר השורות בו יש לרדת לאורך המסך כדי להגיע לנקודה. השורה העליונה היא שורה 0 והשורה התחתונה היא שורה 21. אחר כך מגיע מספר העמודות לרוחב המסך אותו יש לעבור כדי להגיע לנקודה. העמודה השמאלית היא עמודה 0 והעמודה הימנית היא עמודה 31. (בעמ' 80 תוכל לראות את מערך רשת המסך בהפרדה נמוכה). התכנית הבאה ממלאת מקומות תוים אלה בצבעים. מילת המפתח RND (על מקש R) בוחרת צבע דיו אקראי.

## רצועים אקראיים

```
10 BORDER 1: INK RND*7
20 PRINT " ",
30 GO TO 10
```



כאן מופיעים רצועים על פני כל המסך. כדי לגרום להופעת תו בנקודה מסוימת, עליך להשתמש במילת המפתח AT יחד עם PRINT. AT באה אחרי PRINT ומלווה במספר השורה, פסיק, מספר העמודה ונקודה פסיק. הפקודה הבאה:

```
PRINT AT 11,16: "*"
```

לדוגמה, מציגה כוכבית בשורה 11, עמודה 16, שם נמצא מרכז המסך.

## כיצד ליצור צורות קשת בענן

דרך טובה ליצירת צורות צבעוניות היא השימוש בלולאות FOR NEXT בתכניות הגרפיקה. לולאות FOR NEXT הן חלקים בתכנית החוזרים על עצמם מספר מסויים של פעמים. בשורה הפותחת לולאה, אתה יכול להורות למחשב כמה פעמים יש לבצע את הלולאה. כאשר המחשב מבצע זאת, ניתן להשתמש בכך, לדוגמה, להצבת תוים על המסך.

## כיצד לבחור תוים גרפיים

הפוכים. זוהי הדרך להציב תוים גרפיים בשורות תכנית. כדי לצאת מבחירת גרפיקה ולהחזיר את מקשי המספרים למצבם הרגיל, פשוט לחץ שנית על מקש GRAPH.

הרווחים בין אחד לשני. התוים הגרפיים מופיעים בתחתית המסך. החלק הלבן של כל תו במקש הוא צבע הדיו, והחלק השחור צבע הנייר. כעת לחץ על המקשים שנית, יחד עם לחיצת CAPS SHIFT באותו זמן. הפעם יופיעו התוים כאשר צבעי הדיו והנייר

לספקטרום ZX+ יש מערך תוים גרפיים במקלדת המאפשרים תכנות קל של גרפיקה בהפרדה נמוכה. ניתן לראותם על מקשים 1 עד 8. כדי ליצור תוים גרפיים על המסך, לחץ על מקש GRAPH ולחץ על מקשים 1 עד 8, תוך שימוש במקש

## GRAPH

משתמשים במקש זה כדי להעביר את הספקטרום לאופן פעולה - גרפיקה.



## Key 8

משתמשים במקש זה לעיתים קרובות עם GRAPH כדי ליצור ריבוע מלא של צבע.





```

5 BORDER 2: PAPER 1: CLS
10 PRINT INK 2; AT 3, 15, " "
15 PRINT INK 2; AT 4, 15, " "
20 PRINT INK 2; AT 5, 15, " "
30 PRINT INK 5; AT 6, 13, " "
40 FOR L=7 TO 10: PRINT INK 5,
AT L, 13, " " : NEXT L
45 PRINT INK 6; PAPER 0; AT 8, 1
5, "ZX"
50 PRINT INK 2; AT 11, 13, " "
60 FOR L=11 TO 15: PRINT INK 6
; AT L, 13, " " : NEXT L
70 PRINT INK 3; AT 16, 13, " ";
AT 17, " "
80 FOR L=17 TO 21: FOR C=0 TO
31
90 PRINT INK 4; AT L, C, " "
100 NEXT C: NEXT L

```



מילת המפתח TAB המופיעה אחרי PRINT בשורה 70 משמשת למיקום תו לאורך השורה בה עובד המחשב באותו זמן. הקשת TAB המלווה במספר בין 0 ל 31, לקביעת מיקום עמודה.

#### כיצד להשתמש בלולאות FOR NEXT

לולאת FOR NEXT מתחילה תמיד בשורה המכילה את מילות המפתח FOR TO, יחד עם משתנה וערכיו ההתחלתי והסופי. לדוגמה:

```
30 FOR C=1 TO 6
```

כאן המשתנה הוא C. הלולאה אותה מתחילים תכיל שורה(ות) הגורמת למחשב לחזור על הפעולה. שורות התכנית יכולות אף הן להשתמש במשתנה C. לולאות FOR NEXT מסתיימות תמיד במילת המפתח NEXT והמשתנה. לדוגמה:

```
50 NEXT C
```

כאשר התכנית רצתה, כל הלולאה מ-FOR ועד NEXT חוזרת על עצמה מספר פעמים קבוע מראש. המשתנה מתחיל בערך הראשון לפני TO וגדל ב-1 בכל פעם עד שיגיע לגבול אחרי TO. במקרה זה, הלולאה חוזרת 6 פעמים, כאשר C מתחיל בערך 1 והופך להיות 2, 3, 4, 5 ולבסוף 6.

בתכנית הראשונה בעמ' 25, מופיעות 3 לולאות בתוך 'ק'. זה אומר שעבור כל מחזור לולאה 'חיצונית', הלולאה ה'מרכזית' תעבור בכל המחזורים. הלולאה ה'פנימית' מבצעת את מחזוריה בתכיפות רבה, בכל פעם שהלולאה ה'מרכזית' משלימה סיבוב פעולה אחד.

אתה אינך מוגבל לתכנות לולאה אחת בלבד בכל פעם. ניתן להציב לולאה אחת בתוך אחרת, ולרוב עם תוצאות שימושיות מאד. התכנית הבאה תראה לך כיצד שתי לולאות FOR NEXT (אחת 'מקוננת' בתוך האחרת) יכולות לשמש עבור החלפת הצבעים והמקומות שנוצרו על ידי INK ו-AT. אתה יכול לראות כיצד לתכנת לולאות אלה בפיסקה שבתחתית עמוד זה.

#### קשת בענן

```

1 BORDER 0: PAPER 5: CLS
10 LET X=1
20 FOR L=0 TO 21
30 FOR C=1 TO 6
40 PRINT INK C; AT L, C+X, " "
50 NEXT C
60 LET X=X+1
70 NEXT L

```



#### תכנות תמונות

בגרפיקה בהפרדה נמוכה, אתה יכול 'לצייר' תמונות בעזרת חישוב המקומות והצבעים של התווים הגרפיים. ניתן לתכנן את התמונה תוך שימוש ברשת הנקודות בהפרדה נמוכה בעמ' 80. ואז, תוך בחירת התווים הגרפיים בדרך המתוארת בעמוד הבא, הקש פנימה את שורות התכנית אחת אחרי השנייה ליצירת התמונה.

התכנית הבאה מראה את סוג התוצאות שניתן לקבל. את כל הצורות המופיעות אפשר למצוא על מקשי המספרים. אתה יכול לחכות עד הקשת כל המספרים לפני הרצת התכנית, אך אם, במקום זאת, תריץ את התכנית בתום הקשת כל שורה, תוכל לראות כיצד חלקי הרובוט השונים מורכבים יחד. (זכור שאם הקשת תווים גרפיים שגויים, ניתן לערוך אותם בדיוק כמו עריכת מספרים או אותיות שגויות.)



# לוח טיוטה על המסך

הנקודה ההתחלתית היא 0,0 אם לא השתמשנו מקדם ב-PLOT או DRAW בתכנית. אם השתמשנו בהן מקדם, אזי נקודה זו היא נקודת PLOT האחרונה, או הנקודה אליה הגיעו בפקודת DRAW האחרונה, מה שבא אחרון. הצהרת DRAW תצייר עתה את הקו לנקודה החדשה. אם הקו אמור לפנות לשמאל או לתחתית המסך, אזי המרחקים האפקיים והאנכיים מקבלים ערכים שליליים (מינוס). נסה תכנית זאת:

כוכב

```
10 INK 2
20 PLOT 128,174
30 DRAW 70,-140
40 DRAW -152,80
50 DRAW 164,0
60 DRAW -150,-80
70 DRAW 70,140
```



PLOT מזיזה את נקודת ההתחלה אל ראש המסך. אחר כך מציירים את 5 הקווים האדומים בעזרת 5 הצהרות DRAW. כעת הוסף לתכנית את השורות הבאות:

```
4 BORDER 1:PAPER 6:INK 1:CLS
5 CIRCLE 128,87,87
```

הרץ שוב את התכנית, והכוכב האדום יופיע בתוך



עיגול על רקע נייר צבעוני.

CIRCLE דורשת 3 ערכים. שני הראשונים נותנים את מיקום מרכז העיגול, והמספר השלישי נותן את הרדיוס. אתה יכול להוסיף עוד שלישי גם להצהרות

הגרפיקה בספקטרום ZX+ אינה מוגבלת לצורות מגושמות בהפרדה נמוכה. בעזרת יכולת ההפרדה הגבוהה ניתן להשתמש בספקטרום ליצירת תמונות מפורטות עם תוים חדים, קווים ישרים או עקומים חזויים.

גרפיקה בהפרדה גבוהה בנוייה מנקודות רבות הממוקמות בזו אחר זו ליצירת קו או מילוי צורה בצבע. כל נקודה מהווה 1/64 מגודל הריבוע של גרפיקה בהפרדה נמוכה. אם תקיש את הפקודה:

PLOT 128,87

כעת תראה נקודה כזאת במרכז המסך. הנקודות המשמשות בגרפיקה בהפרדה גבוהה נקראות פיקסלים, בקיצור מאנגלית ל"תאי תמונה". בדומה לתו בהפרדה נמוכה, כל תא תמונה דורש שני מספרים להגדרת מיקומו. אלה אינם אותם המספרים המשמשים בתצוגות בהפרדה נמוכה.

## רשת הנקודות בהפרדה גבוהה

רשת הנקודות בהפרדה גבוהה בנוייה מ-256 תאי תמונה לרוחב המסך 176 לאורכו. ברם, שלא כמו בתצוגות בהפרדה נמוכה, המספר הראשון הוא הקואורדינטה האפקית - המיקום לרוחב המסך. מספרי נקודות אלה נעים מ-0 בפינה השמאלית עד 255 בפינה הימנית. המספר השני הוא הקואורדינטה האנכית, אך כאן המספרים נעים בין 0 בתחתית ל-175 בראש המסך. נקודה 0,0 היא הפינה השמאלית התחתונה במסך, ולא הפינה השמאלית העליונה כמו בהפרדה נמוכה. ראה טבלת רשת נקודות בהפרדה גבוהה בעמ' 80.

## התייה וציור

אתה זקוק לשלוש מילות מפתח בלבד ליצירת גרפיקה בהפרדה גבוהה - DRAW, PLOT, CIRCLE. מלווה בקואורדינטות האפקיות והאנכיות כשהן מופרדות בפסיק, ואז ממוקם תא תמונה בנקודה זאת. DRAW מלווה אף היא בשני מספרים המופרדים בפסיק, אך אלה אינם קואורדינטות הנקודה. במקום זאת, אלה הם המרחקים בפיקסלים מנקודה אחת לנקודה אחרת לאורך ולרוחב המסך, ואז DRAW מציירת קו בין שתי הנקודות.



## לוח טיוטה על המסך

הספקטרום גמיש מאד. אינך צריך לכתוב תכנית כל אימת שתדע ליצור תמונה או צורה. במקום זאת, אתה יכול להשתמש בתכנית המאפשרת לך לבנות תמונה ישירות על המסך. להלן מופיעה תכנית פשוטה מאד המאפשרת זאת.

התכנית פותחת במילת המפתח INPUT לקבלת מספר דיו. אחר כך, תוך שימוש נוסף ב-INPUT (הפעם גם סימן & לסימון מחרוזת), היא גורמת למחשב לצייר קווים קצרים כל אימת שתלחץ על 4 מקשים מוגדרים - u, d, l, r.

## טיוטה ודוגמה

```

10 INPUT "INK " : i
20 BORDER 3: PAPER 0: INK i: C
LS
30 PLOT 25,25
40 LET X=5
50 INPUT K$
60 IF K$="u" THEN DRAW 0,X
70 IF K$="d" THEN DRAW 0,-X
80 IF K$="r" THEN DRAW X,0
90 IF K$="l" THEN DRAW -X,0
100 GO TO 50

```



## תכנית הדוגמה עם IF THEN

שורות 60 עד 90 בתכנית הטיוטה מכילות הצהרות IF THEN אלה מאפשרות לספקטרום לקבל החלטות. במקרה זה, המחשב בודק אם המקש עליו לחצת הוא u, d, l או r. אם אחד מאלה נלחץ, אזי המחשב מורה על ציור קו. הוא לא יצייר קו אם תוקש אות גדילה.

IF מלווה תמיד בביטוי שהספקטרום בודק האם הוא אמת או שהוא קיים - כמו מקשים מסוימים שנלחצו. אם הביטוי אמת או קיים, אזי תתבצע הפעולה הבאה אחרי THEN. אם לא, אזי התכנית עוברת לשורה הבאה. כל מה שבא אחרי THEN בשורה זו כפוף להתלטה בשורה שלהלן.

```
110 IF b=5 THEN PRINT"*":GOTO 200
```

המחשב יפנה לשורה 200 רק אם b הוא 5.

DRAW. נסה ערכים בין 2 ל-27 עם התכנית וראה מה קורה.

## כיצד למלא צורות

אתה יכול ליצור בקלות צורות מלאות בהפרדה גבוהה על ציור קווים רבים כשהם קרובים אחד לשני. הדבר ניתן להיעשות בלולאת FOR NEXT אשר משנה את נקודות המיקום ב-DRAW כך שהן עולות ב-1 בכל פעם.

## משולש מלא

```

10 BORDER 1: PAPER 6: INK 2: C
LS
20 FOR X=-100 TO 100
30 PLOT 120,150
40 DRAW X,-120
50 NEXT X

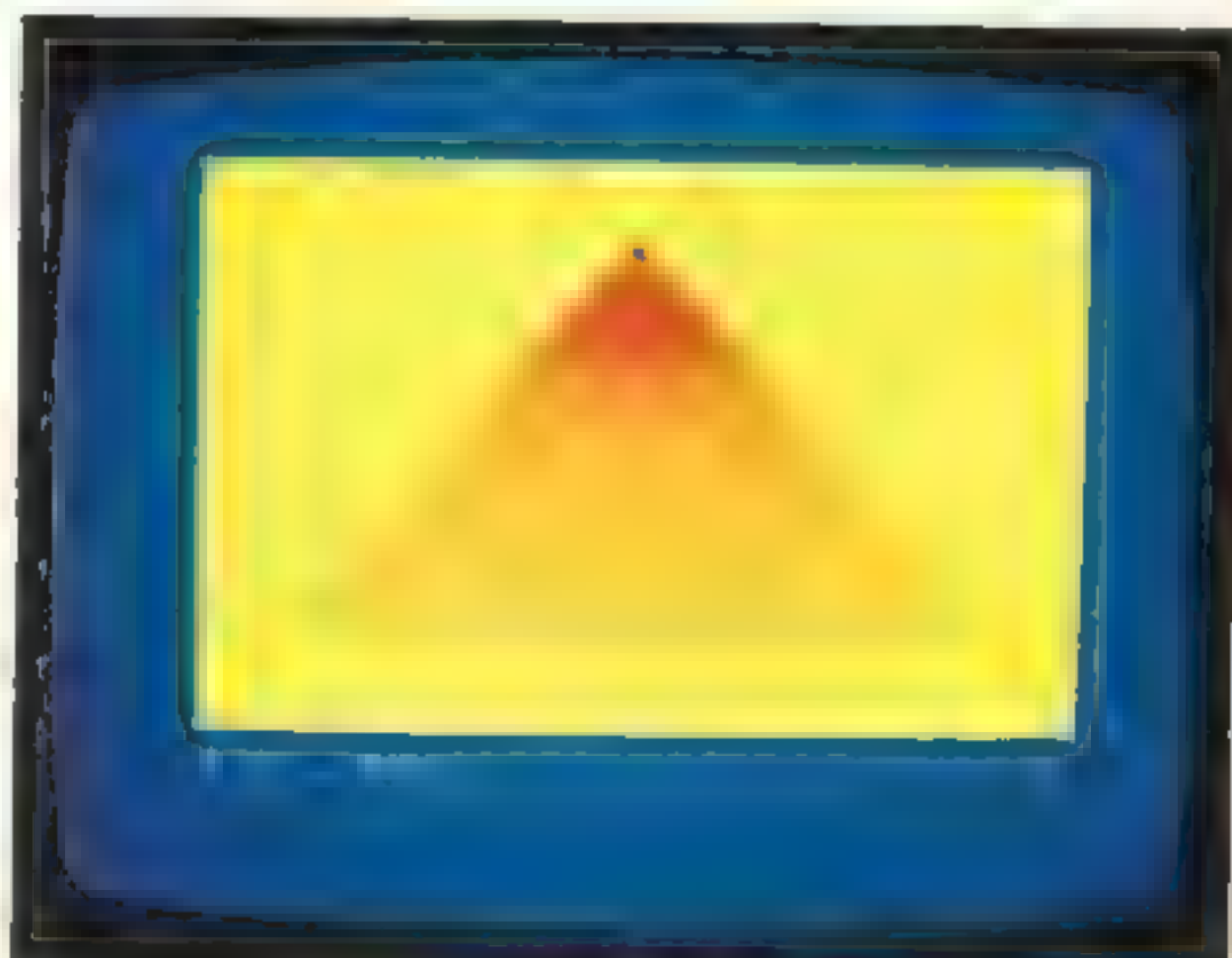
```



אתה יכול לקבל אפקט מעניין אם תצייר קווים מרווחים מעט זה מזה. ניתן לעשות זאת בהוספת מילת המפתח STEP ומספר להצהרת FOR. בטכניקה זו משתמשים בתכנית "הזריחה המונצצת" בעמ' 11, אך היא תפעל על תכנית המשולש באותה צורה. הקש שורה 20 אחרת והרץ שנית את התכנית.

```
20 FOR X=-100 TO 100 STEP 4
```

הפעם תופיע הצורה דמויית-מניפה המופיעה להלן. הסיבה לכך היא ש-STEP גורמת ל-X לעלות בקפיצות של 4 במקום להגדילו ב-1 בכל פעם שמציירים קו.





# תכנן בעצמך צורות ותמונות שונות

אתה יכול ליצור בעצמך צורות ותמונות שונות בספקטרום ZX+, תוך שימוש בגרפיקה בהפרדה נמוכה, גרפיקה בהפרדה גבוהה, או שתיהן יחד. הדרך הטובה ביותר להתמודד עם גרפיקה היא להתוות את הצורה המבוקשת על העתק רשת הנקודות בעמ' 80. אחר כך בנה את התכנית שתיצור קוים וצורות במקומות הנכונים על המסך.

כדי לצייר צורות ותמונות, ניתן להשתמש לרוב בלולאות FOR NEXT החוזרות על קטע תכנית מספר פעמים מוגדר מראש. בכל פעם משתנים צבעי ומיקומי התוים או השורות, לרוב באופן קבוע. להלן מופיעה תכנית המשתמשת בטכניקה זאת.

ריבועים

```
10 BORDER 0: PAPER 0: CLS
20 FOR x=7 TO 0 STEP -1
30 INK x
40 FOR l=11-x TO 11+x
50 FOR c=16-x TO 16+x
60 PRINT AT l,c: " "
70 NEXT c
80 NEXT l
90 NEXT x
```



תכנית זו מכילה 3 לולאות FOR NEXT. לולאת ה-x משנה את צבע וגודל הריבועים הגדולים, בעוד שלולאות l ו-c משנות את מיקום השורה והעמודה של הריבוע הקטן בכל פעם שהוא מודפס. נסה לשנות את הריבוע בשורה 60 לכוכב או תו אחר כלשהו במקלדת.

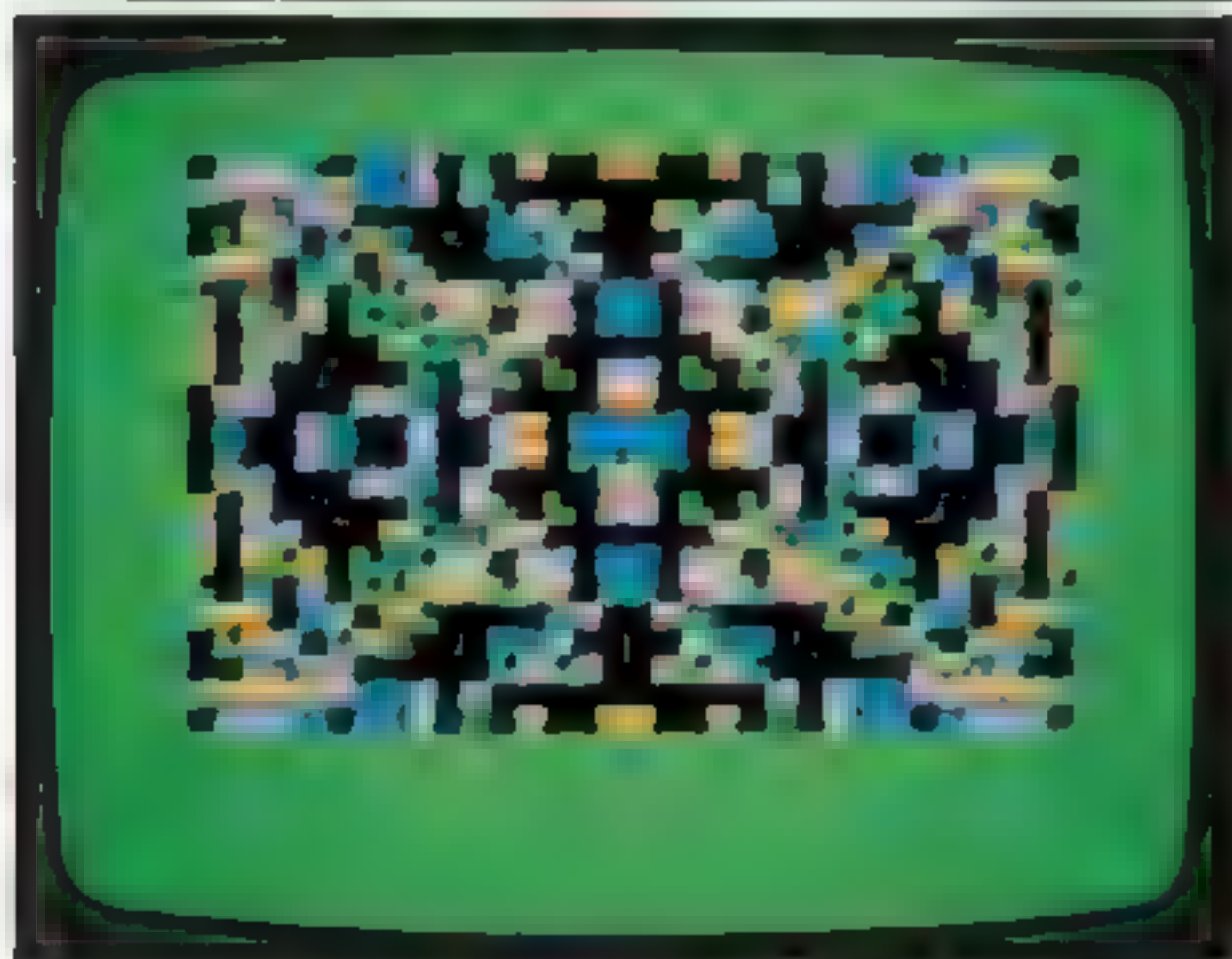
אפקטים אקראיים ושגרות

השימוש בלולאות אינו נותן בהכרח צורות זהות בכל פעם שמורצת תכנית גרפית. ע"י הכנסת מילת המפתח RND (קיצור ל-Random - אקראי) בלולאות, ניתן ליצור צבעים, מיקומים ותכונות תצוגה אחרות המשתנות בכל פעם. התבונן בתכנית המוזאיקה בעמ' 10. התכנית פועלת משום שצבע

הדיו הוא  $RND * 7$ , שמשמעותו כל מספר עם נקודה עשרונית בתחום שבין 0 עד 7. INK משנה זאת למספר השלם הקרוב ביותר. לכן בכל פעם שמוצג ריבוע, צבעו יכול להיות כל צבע שבין INK ו-7. התכנית הבאה מציירת צורות סימטריות של תוים גרפיים על המסך. נעשה כאן שימוש ב-RND לשינוי התוים ומיקומם. המשתנים l ו-c נותנים את צבעי הדיו והנייר, ר-c מציין כמה צורות יוצגו (במקרה זה ארבע). המשתנה n נותן את מספר התוים בכל צורה, בעוד ש-r הוא מספר אקראי בתחום 129 עד 142. הצהרת GOSUB 1000 בשורה 50 שולחת את המחשב אל שגרה.

צורות סימטריות

```
10 BORDER 4: PAPER 0: CLS
20 LET i=4: LET p=0
30 FOR s=1 TO 4
40 LET x=RND*13+129
50 FOR n=1 TO 40: GO SUB 1000:
NEXT n
60 LET i=i+1: LET p=p+1
70 PAUSE 100
80 NEXT s
90 STOP
1000 LET l=INT (RND*11)
1010 LET c=INT (RND*16)
1020 INK i: PAPER p
1030 PRINT AT l,c:CHR$(x)
1040 PRINT AT l,31-c:CHR$(x)
1050 PRINT AT 21-l,c:CHR$(x)
1060 PRINT AT 21-l,31-c:CHR$(x)
1070 BEEP 0.01,(l+c)/3
1080 RETURN
```



שגרה היא קבוצת שורות המתנהגת כתכנית בתוך תכנית. בתכנית זאת השגרה נמצאת בשורה 1000. היא מציגה תו גרפי בארבעת רבעי המסך כך שכל אחד מרוחק במידה שווה מהמרכז (נקודה 11,16). מרחק זה נתון בשורות 1000 ו-1010. נותן את המרחב בשורות 1020 את המרחק בעמודות. INT משנה את המספר האקראי למספר שלם כך ש-r הוא מספר שלם כלשהו בין 0 ל-142. שורות 1030 עד 1060 מציגות את התו הגרפי בעל קוד x (ראה טבלת מערך תוים בעמ' 51). BEEP מייצרת צליל שגבהו קשור למיקום הנקודה, ואז RETURN בשורה 1080 שולח את התכנית חזרה להצהרה הבאה אחרי GOSUB בשורה 50.



```

120 LET x=RND*255
130 LET y=RND*104+71
140 LET l=INT (175-y)/8
150 LET c=INT (x/8)
160 PLOT 0,0: DRAW OVER 1;x,y
170 BEEP 0.01,x/4
180 PLOT 0,0: DRAW OVER 1;x,y
190 PRINT AT l,c;"*"
200 GO TO 120

```



1 OVER בשורות 160 עד 180 מאפשר לשורה הראשונה לצייר את קו הלייזר ולשורה השנייה למחוק אותה ללא שינוי יתר התמונה. שמור תכנית זו מאחר ותזדקק לה במועד מאוחר יותר.

#### INVERSE, BRIGHT, FLASH

שלוש מילות מפתח אלה מאפשרות גמישות רבה בהפעלת צבעי הספקטרום. כל מילת מפתח מלווה ב-0 או 1, ועליך להציבם בהצהרות PRINT בתנאי שיש נקודה פסיק אחרי ה-0 או ה-1. ה-1 FLASH גורמת למקומות תווים להבהב בין צבעי הדיו והנייר, בעוד ש-0 BRIGHT יוצרת צבעים בהירים יותר. ה-1 INVERSE משנה את צבע הדיו לצבע הנייר ולהיפך. הצבת 0 אחרי מילות מפתח אלה מחזירה את התצוגה למצב הקודם. נסה להכניס שינויים אלה בתכנות בפרק זה כדי לראות את פעולת מילות המפתח. בתכנית הריבועים, שנה את הריבוע בשורה 60 לכוכב ואז הוסף

#### 15 INVERSE 1

כעת הכוכבים יופיעו בשחור (צבע הנייר) כנגד פסים צבעוניים (צבעי הדיו המתחלפים). הקש 0 INVERSE לפני שתמשיך.

בתכנית הצורות הסימטריות, הוסף את השורות הבאות כדי לראות את פעולת BRIGHT ו-FLASH

#### 15 BRIGHT 1 16 FLASH 1

שים לב כיצד FLASH גורם לצורה להיראות כאילו היא נעה, הקש 0:CLS FLASH להמסקת הבהוב התצוגה. כל השינויים האלה משפיעים על התצוגה המלאה שיוצרת כל תכנית השימוש ב-BRIGHT, FLASH או INVERSE בתוך הצהרת PRINT מגביל את פעולת 3 מילות המפתח להדפסה הנוצרת בשורה זו בלבד.

שורה 60 משנה את צבעי הדיו והנייר, ואז

100 PAUSE בשורה 70 משהה את התכנית למשך 2 שניות לפני השלמת הלולאה וחזרה לנקודת ההתחלה. בשורה 90 נדרש STOP לעצירת התכנית לבל תמשיך אל תוך השגרה לאחר הלולאה הרביעית.

אתה יכול לשנות תכנית זו ע"י שינוי בערך 4

בשורה 30, 40 בשורה 50 למספרים אחרים. אם

תגדיל יותר את טווח x בשורה 40, תקבל תווים

אחרים שיופיעו על המסך. אל תגדיל את טווח ערכי i

רק ליותר מאשר 7.

#### השימוש בלולאות FOR NEXT בגרפיקה

ניתן להשתמש ביעילות בלולאות FOR NEXT בגרפיקה בהפרדה גבוהה ליצירת תמונות הבנויות מקווים וצורות קבועים. הקש פנימה והרץ את התכנית הבאה. על ידי שימוש ב-PLOT ו-DRAW בלבד, שתי לולאות FOR NEXT מציירות בתחילה קווים על הקרקע ואחר כך 5 משולשים או פירמידות מלאות.

#### פירמידות

```

10 BORDER 0: PAPER 1: INK 0
20 CLS
30 FOR y=0 TO 20 STEP 2
40 PLOT 0,y
50 DRAW 255,0
60 NEXT y
70 FOR n=100 TO 220 STEP 30
80 FOR x=-10-n/10 TO 10+n/10
90 PLOT n,35+n/10
100 DRAW x,-n/4
110 NEXT x: NEXT n

```



כעת הוסף את השורות בתכנית הבאה. כאשר תריץ אותה שנית, תמצא שקו לייזר נורית ללא הרף לשמי הלילה, ויוצרת צורות כוכבים. הקו מצוירת מפינת המסך לנקודה x,y, כאשר x ו-y הם מספרים אקראיים. מספרים אלה מומרים למספרי מיקום כוכבים בהפרדה נמוכה.



# כיצד ליצור תוים משלך במחשב

```

10 POKE USR "S",BIN 00111100
11 POKE USR "S"+1,BIN 01111110
12 POKE USR "S"+2,BIN 11011011
13 POKE USR "S"+3,BIN 11111111
14 POKE USR "S"+4,BIN 10111101
15 POKE USR "S"+5,BIN 10100101
16 POKE USR "S"+6,BIN 10100101
17 POKE USR "S"+7,BIN 00100100

```

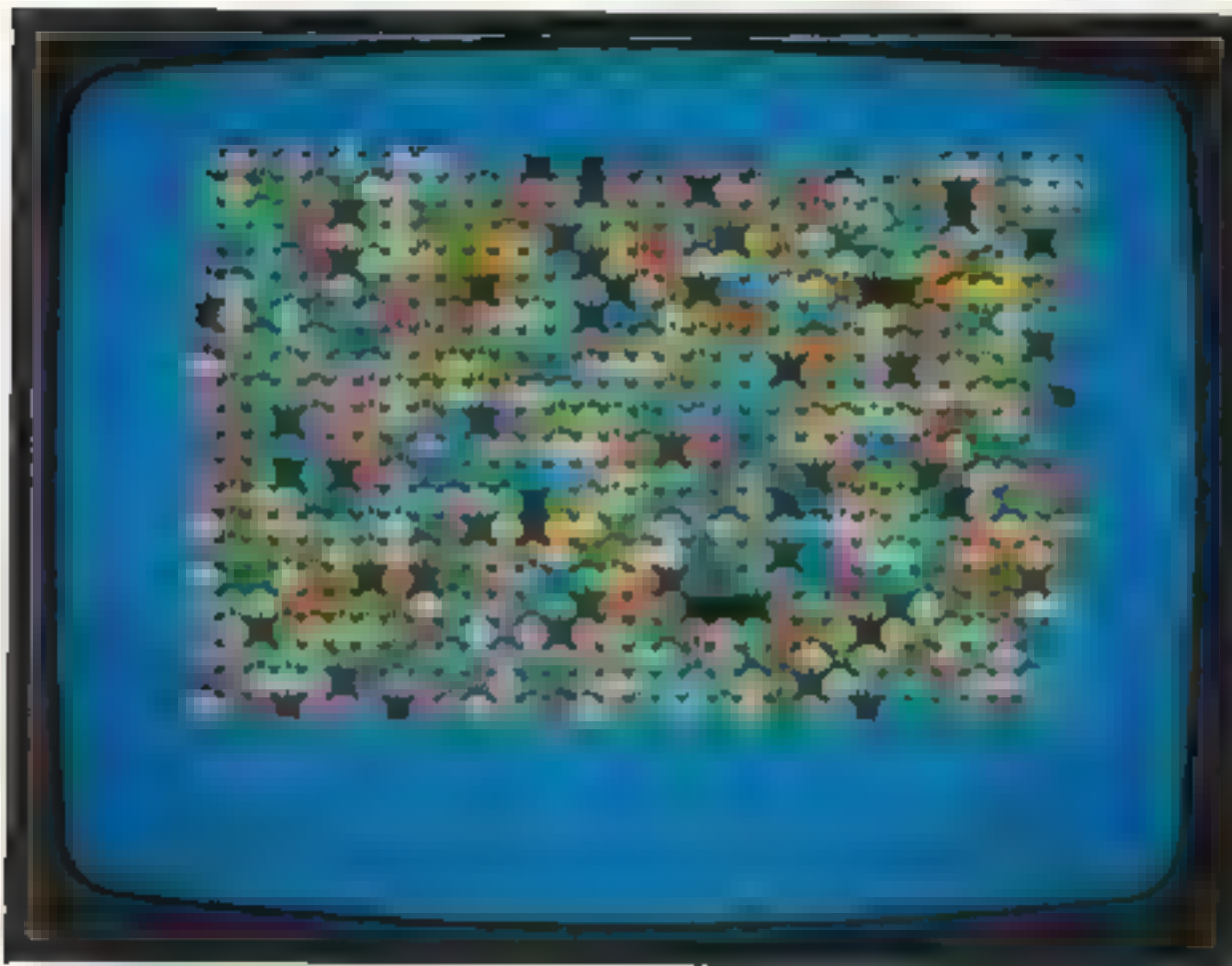
כעת הרץ את התכנית ולחץ GRAPH R, מופיע העכביש! עתה הוסף את השורות הבאות, לקבלת העכביש בשורה 30 בדרך זו, והרץ את התכנית. עכבישים יופיעו על פני כל המסך.

## עכבישים

```

20 BORDER 1: PAPER 0:CLS
30 PRINT INK AND+7," ";
40 GO TO 30

```



כאשר אתה מעצב תוים משלך, זכור שלא תוכל לראותם על המסך אלא אם הרצת תכנית אשר מגדירה אותם. עד אז, הם יופיעו ברישומים כאותיות.

## כיצד לערבב צבעים עם ריבועים מנומרים

אתה יכול לדמות בקלות ערבוב צבעים בספקטרום. כדי לעשות זאת תצטרך ליצור תו שכאשר יודפס יתן רבוע שהוא 50% בצבע דיו ר 50% בצבע הנייר.

```

10 FOR X=0 TO 5 STEP 2
20 POKE USR "2"+X,BIN 10101010
30 POKE USR "2"+X+1,BIN 01010101
40 NEXT X

```

עליך רק להגדיר שתי שורות פיקסלים ולהורות למחשב להשתמש בהן לחילופין בתוך תו.

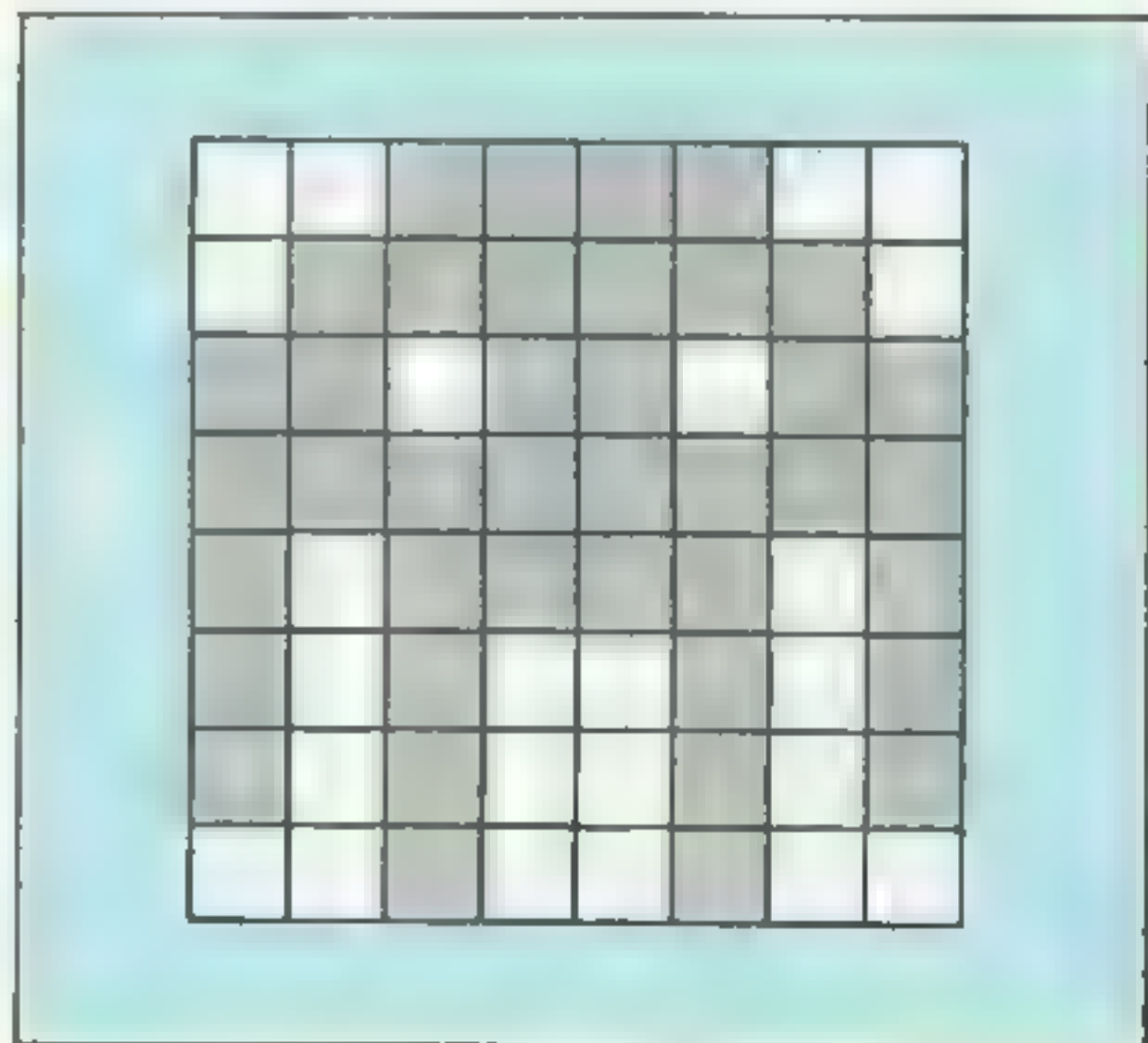
כאשר תריץ את התכנית, תראה רבוע מנומר. אם תנקוט באותה טכניקה בתכנית המכילה מילות מפתח לצבעים, הרבוע המנומר יצור צבע שיהיה תערובת של צבעי הדיו והנייר בתכנית.

הספקטרום ZX+ אינו מוגבל רק לתוים גרפיים שניתן להקיש מהמקלדת. בחלק מיוחד של הזכרון, ניתן לשמור תוים אחרים שתכנת בעצמך. אלה נקראים תוים גרפיים מוגדרים משתמש וכל תכנית יכולה להכיל עד 21 תוים כאלה.

כל תו בנוי מעד 64 נקודות קטנות או פיקסלים בצבע דיו. הפיקסלים מסודרים ב-8 שורות בנות 8 פיקסלים כל אחת, וכל תו תופס מקום של תו אחד ברשת הנקודות של הפרדה נמוכה – בדיוק כמו התוים הגרפיים התקניים במקלדת.

## עיצוב תו גרפי

ראשית צייר רשת 8x8 כמתואר להלן. מלא בצבע חלק מהריבועים ליצירת תו. רבועים אלה מייצגים את הפיקסלים בצבע דיו. כל רבוע מלא נחשב כערך 1 וכל רבוע ריק נחשב כ-0. התכנית הבאה מתארת בניית עכביש.



כל אחד מהתוים המוגדרים ע"י המשתמש מזהה ע"י אות מ"א עד U (או A עד U – אין הבדל). לתיכנות האות, הקש 8 הצהרות POKE USR כשכל אחת מהן מסתיימת ב"BIN בלוי מספר בינארי המורכב משמונת ה"ס או 1 בכל שורה של רשת הנקודות. הבה ונקרא את תו העכביש S.



## ציור לוח שחמט

כאן מוצעת תכנית המציגה לוח שחמט על המסך ועורכת עליו את הכלים לקראת משחק. אתה יכול ליצור צבעים שיופיעו ברשימה בעזרת קודי בקרת צבע - ראה את הטבלה שלהלן.

## לוח שחמט

```

10 FOR x=1 TO 6
20 READ :$
30 GO SUB 500
40 NEXT x
50 BORDER 4: PAPER 1: CLS
60 FOR l=7 TO 14 STEP 2: FOR c
=10 TO 19 STEP 2
70 PRINT AT l,c;" ";AT l+1,c;
80 NEXT c NEXT l
90 PRINT AT 7,12,"KING"
100 PRINT AT 8,12,"QUEEN"
110 PRINT AT 13,12,"PAWN"
120 PRINT AT 14,12,"PAWN"
130 GO TO 130
140 FOR n=0 TO 7
150 READ y
160 POKE USA :$+n,y
170 NEXT n
180 RETURN

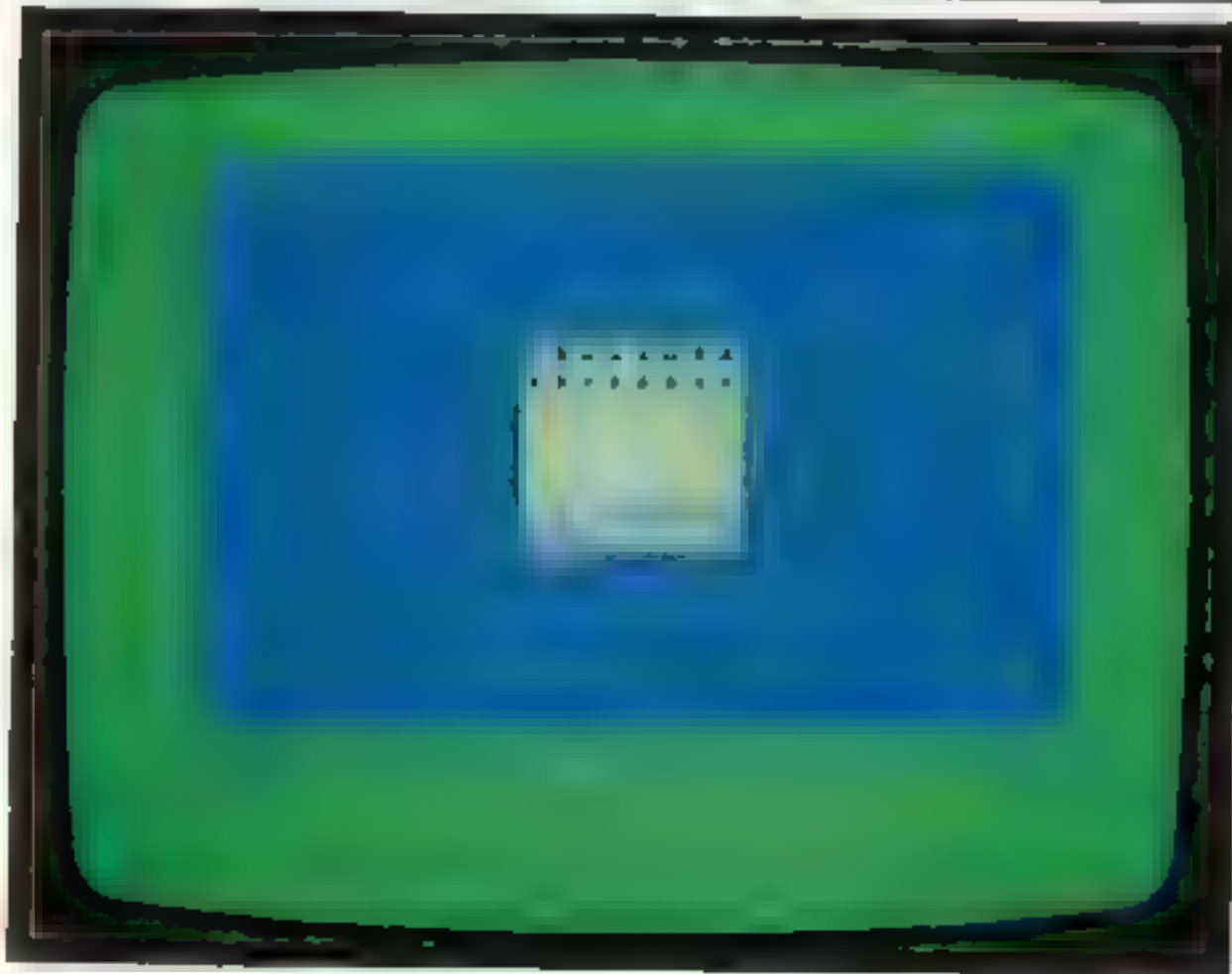
```

הכלים מוגדרים בשורה המתחילה בשורה 500.

```

550 DATA "p",0,0,16,56,56,16,12
560 DATA "r",0,34,124,56,56,124,0
570 DATA "n",0,16,56,120,24,56,124,0
580 DATA "b",0,16,40,60,100,56,124,0
590 DATA "k",0,16,56,16,56,60,5,0
600 DATA "q",0,34,40,16,100,124,124,0

```



## פישוט הגרפיקה בעזרת READ DATA

להלן מוצעת דרך קלה יותר ליצירת גרפיקה במחשב בעזרת שימוש במספרים עשרוניים בפקודות READ DATA. ראשית שנה את 8 המספרים הבינאריים הבנויים מ-0 ו-1 למספרים עשרוניים. עשה זאת בעזרת הקשת PRINT BIN המלווה במספר. לדוגמה:

PRINT BIN 00111100

הספקטרום יציג 60, שהוא אקוויולנטי למספר הבינארי 00111100. במקרה של העכביש, 8 המספרים העשרוניים הם 60, 126, 219, 255, 165, 165, 36. כעת אתה יכול להשתמש ב-READ DATA. שתי מילות מפתח אלה נותנות דרך נוחה להזנת ערכים רבים כמו מספרים למשתנים בתכנית. READ מלווה במשתנה - אות אחת או יותר אם עוסקים במספרים, או אות בודדת בלויית \$ במקרה של מחרוזות.

מכיוון שכאן תרצה לקרוא מספרים עשרוניים, בחר במשתנה מספרי שיקרא y.

כאשר הספקטרום נתקל במלה READ, הוא פונה להצהרת DATA הראשונה בתכנית. הצהרה זאת כוללת רשימת ערכים המופרדים בפסיקים. המחשב לוקח את הערך הראשון ברשימה ונותן אותו למשתנה הבא אחרי READ. הערך השני ינתן למשתנה הבא, וכן הלאה לפי הסדר. להלן מופיעה התכנית החדשה ליצירת העכביש.

```

10 FOR x=0 TO 7
20 READ y
30 POKE USA :$+x,y
40 NEXT x
50 DATA 60,126,219,255,165,165,36,165

```

התכנית תאחסן למעשה בזכרון 8 מספרים עשרוניים כלשהם ליצירת תו. עליך רק לשנות את \$ בשורה 30 לאות שתדע, ואחרי DATA בשורה 50 תקיש את 8 המספרים, כשהם מופרדים בפסיק. לחץ GRAPH ואת האות לקבלת התו לאחר שהוצאת את התכנית.

## טבלת שימוש בקודי בקרת צבע

EXTEND MODE ואח"כ על מקש מספר עם יו בלי CAPS SHIFT. טבלה זו מראה כיצד תוכל לבחור כל אחד מהצבעים.

לתצוגה יחליפו את צבעיהם ברישום עצמו ויופיעו בצבעים אלה גם על המסך. כדי להגיע לקודים, לחץ תחילה על

במקום להשתמש במילות מפתח כמו INK PAPER, ניתן להציב קודי בקרה בתוך הצהרות PRINT אחרי סימן המרכאות הראשון. התוים המיועדים

שחור	הבהוב	הבהוב לא פועל	לבן	צהוב	תכלת	ירוק	סגול	אדום	כחול
שחור	הבהוב	הבהוב לא פועל	לבן	צהוב	תכלת	ירוק	סגול	אדום	כחול
שחור	הבהוב	הבהוב לא פועל	לבן	צהוב	תכלת	ירוק	סגול	אדום	כחול



כעת הוסף שורות אלה לתכנית והרץ אותה שנית.  
תוכל לראות הנפשה אך הפעם בכוח שונה.

### עכביש מתרוצץ

```
150 FOR c=3 TO 30
160 PRINT AT 21,c;" "
170 PRINT AT 21,c+1; INK 2;"A"
180 NEXT c
```



כעת העכביש מתרוצץ הצידה ברגע שהגיע ארצה.  
השורות הנוספות יוצרות לולאת FOR NEXT נוספת  
המבקרת את מיקום העמודה c. שים לב לכך שקודם  
מודפס רווח ואז מוצג העכביש במיקום העמודה  
הבאה. כך נוצר מצב בו העכביש נעלם מנקודה אחת  
ומופיע בנקודה הבאה בכיוון ימין. תמיד עדיף למחוק  
תו לפני הדפסתו במיקום הבא, מאשר להדפיס קודם  
את התו במיקום החדש ורק אז למחוק מהמיקום  
הישן. כך מונעים או מפחיתים הבהוב בגרפיקה נעה.

### אימון קליעה למטרה

במשחקי מחשב רבים, מתרחשת לרוב פעולה כאשר  
שני גופים נעים מתנגשים או שגוף נחבט בקיר. כיצד  
יודע המחשב מתי אמורה להתרחש התפוצצות או  
התנגשות?

זיהוי התנגשויות אינו קשה. אם שני תווים  
מודפסים בנקודה l,c (עבור שורה ועמודה) ומיקום  
v,h (עבור מיקום אנכי ואפקי) אזי אם  $l = v$  ו  $h = c$ ,  
שניהם חייבים להיות באותו מקום. ניתן לכתוב זאת  
כהצהרה, לדוגמה:

```
160 IF l=v AND c=h THEN PRINT "CRASH!"
```

דרך אחרת לזיהוי התנגשויות היא בעזרת צבע. מחק  
את תכנית העכביש בהקשת NEW. הקש פנימה את  
תכנית הפירמידות בעמ' 31 במלואה, או טען את  
התכנית אם שמרת אותה על סרט. אתה יכול עתה  
לשפרה ולשלבנה עם העכביש (הנמצא עדיין בזכרון  
אלא אם כבית את המחשב או לחצת RESET) ליצירת  
תכנית אחרת.

## הנפשה

גרפיקה במחשב נראית במיטבה כאשר התווים או  
השורות נעים על המסך, ויצירת הנפשה בספקטרום  
אינה קשה. כל מה שעליך לעשות הוא להמשיך  
ולשנות את מיקום הדפסת תו או ציור שורה. הדרך  
הטובה ביותר לצורך זה היא בעזרת לולאה אחת או  
יותר של FOR NEXT

### תנועה אנכית ואפקית

הקש פנימה והרץ תכנית זאת. אם לא כבית את  
הספקטרום או לחצת על כפתור RESET מאז יצירת  
תו העכביש בעמ' 32, אל תטרח להקיש שורות 10 עד  
50. התו הגרפי ישמר עדיין בזכרון תחת "S".

### עכביש נופל

```
5 BORDER 3: PAPER 5: CLS
10 FOR x=0 TO 7
20 READ y
30 POKE USA "s"+x,y
40 NEXT x
50 DATA 60,126,219,255,169,165
165,36
60 FOR x=0 TO 7
70 READ y
80 POKE USA "t"+x,y
90 NEXT x
100 DATA 16,16,16,16,16,16,16,1
6
110 FOR l=0 TO 20
120 PRINT AT l,3; INK 0;"|"
130 PRINT AT l+1,3; INK 2;"A"
140 NEXT l
```

בכל פעם שתריץ את התכנית תוכל לראות את  
העכביש נופל לאורך המסך כשהוא תלוי על קוריו.  
שורות 60 עד 100 בתכנית יוצרות תו גרפי אחר  
( "t" ) עבור הקור. ההנפשה מתבצעת בשורות 110 עד  
140, היוצרות לולאת FOR NEXT שבה l (מספר  
השורה) משתנה מ-0 עד 20. בכל פעם שהלולאה  
חוזרת על עצמה, אורך קור מודפס בנקודה אחת  
והעכביש מודפס בנקודה הבאה מתחתיו. בפעם  
הבאה, העכביש מוחלף באורך קור נוסף והעכביש  
מופיע שוב מתחתיו. בצורה זו, העכביש משתלשל  
מטה במהירות לאורך הקור עד שהוא מגיע לשורה  
21, שהיא תחתית שטח התצוגה.

הספקטרום מסוגל לחשב את מיקום הנקודות  
הבאות במהירות רבה, כך שהעכביש צונח במהירות.  
כדי להאיט את תנועתו, הכנס שורה זאת:

```
135 PAUSE 10
```

פקודה זו גורמת למחשב להמתין חמישית שנייה בכל  
פעם לפני הדפסת העכביש בנקודה הבאה. נסה  
לשנות 10 לערכים אחרים וראה כיצד משתנה  
המהירות.



כאשר קץ הלייזר פוגעת בעכבישים, הם הופכים לרגע לצהובים. כאשר הקו הנוצר ע"י DRAW חוזר למיקום תו העכביש, צבע הדיו משתנה לצבע השורה, שהוא צהוב. בשורה 190, ATTR מגלה אם העכביש מצהיב ושולח את המחשב לשיגרת ההתפוצצות הנמצאת בשורה 500.

#### הקפצת כדור

תכניות גרפיות רבות מתארות צורות המנתרות מקצות המסך. תכנית זו מראה לך כיצד הדבר מתבצע. המשתנים  $v$  ו- $h$  פועלים באותה צורה כמו תכנית העכבישים המתפוצצים, אך כאן נוסף  $+1$  או  $-1$  ל- $v$  או  $h$  כדי להניע את הכדור מעלה או מטה, ושמאלה או ימינה.  $SCREEN$  בודק האם קיים  $x$  בנקודה  $v, h$ :

#### כדור מנתר

```
10 BORDER 1
20 FOR Z=1 TO 10
30 LET H=INT (RND*25): LET V=I
NT (RND*21)
40 PRINT INK 2; PAPER 6; FLASH
1, AT V, H; "X"
50 NEXT Z
60 LET X=1: LET Y=1
70 PRINT AT V, H; " "
80 LET V=V+Y: LET H=H+X
90 IF H=0 OR H=31 THEN LET X=-
X: BEEP .2, 24
100 IF V=0 OR V=21 THEN LET Y=-
Y: BEEP .2, 12
110 IF SCREEN (V, H) = "X" THEN P
RINT INK 1; PAPER 5; AT V, H; "!"
120 PRINT AT V, H; "O"
130 PAUSE 2
140 GO TO 70
```



#### שימוש במאפיינים

מילת המפתח ATTR מגלה את ה"מאפיינים" בנקודה מסוימת על המסך. המאפיינים הם צבעי הדיו והנייר והאם הנקודה מהבהבת או בהירה. בתכנית העכבישים המתפוצצים, ATTR מבטיחה שהעכביש מושמד אם הוא מצהיב. זהו, איפוא, צבע הדיו שלו (מספר 6). צבע הנייר הוא כחול (מספר 1) והעכביש אינו מהבהב או בהיר. זה אומר שסהל' המאפיינים הוא 14. ההסבר על ATTR במדריך העיון למתכנת מתאר כיצד מגיעים למספרים אלה.

ראשית הוסף את השורות הבאות, היוצרות גרפיקת התפוצצות הנקראת "e".

```
5 FOR X=0 TO 7
6 READ Y
7 POKE USR "e"+X, Y
8 NEXT X
9 DATA 145, 82, 44, 121, 150, 52, 7
4, 137
```

נעת מחק שורה 190 והוסף או שנה את השורות הבאות.

#### עכבישים ופירמידות

```
114 LET H=RND*31
115 FOR V=0 TO 20
117 PRINT AT V, H; " " AT V+1, H;
INK 4; "S"
200 NEXT V
205 PRINT AT 21, H; FLASH 1; INK
2; PAPER 5; "S"
210 GO TO 114
```



צורות הכוכבים לא מופיעים עוד. במקומם נופלים מהשמיים עכבישים המכרסמים את הפירמידות והקרקע. מה שביצעת הוא הוספת לולאת FOR NEXT אשר בה  $v$  ו- $h$  נותנים את מיקום העכביש. המשתנה  $h$  הוא אקראי, כך שהעכבישים מתחילים את צניחתם האנכית במקומות שונים על המסך. נעת הוסף שורות אלה.

#### עכבישים מתפוצצים

```
190 IF ATTR (V+1, H) = 14 THEN GO
TO 500
500 PRINT AT V+1, H; FLASH 1; PA
PER 2; "E"
510 PAUSE 100
520 GO TO 114
```





# כיצד ליצור מוסיקה ואפקטים קוליים

לספקטרום ZX+ יש סינתסייזר קולי היכול לעורר את תכניותיך לחיים עם מגוון גדול של קולות מוסיקליים ואפקטים קוליים מיוחדים. הוא פשוט לשימוש, אפילו אם אין לך ידע רחב במוסיקה. הסינתסייזר מפיק אות קולי המועבר לרמקול הפנימי בספקטרום.

## תיכנות קולות

כדי להפיק צליל בספקטרום, עליך להשתמש במילת מפתח אחת בלבד – BEEP. היא מלווה בשני מספרים או משתנים המייצגים מספרים. הראשון מורה למחשב מהו משך (בשניות) הצליל, והשני מודיע מהו גובה הצליל. גובה הצליל נמדד ביחידות חצאי טונים. ערכי הגובה הם 0 עבור דו אמצעי, 1 עבור דו דיאז, 1 – עבור סי נמוך (C<sub>6</sub>) וכן הלאה. כדי לשמוע את טווח הצלילים המלא שהספקטרום מפיק, הרץ תכנית זאת.

הספקטרום עובר דרך כל טווח הטונים מהצליל הגבוה ביותר (69) ועד הנמוך (60 –), תמצא שהצלילים הגבוהים ביותר בקושי נשמעים ואילו הנמוכים נשמעים כנקישות. הסיבה לכך היא שצלילים אלה נמצאים על סף יכולת השמיעה של אוזן האדם.

```
10 FOR P=60 TO -60 STEP -1
20 BEEP 0.2,P
30 PRINT AT 0.2 " ";AT 0,P
40 NEXT P
```

הטבלה שלהלן מראה את ערכי גובה הצלילים כך שבעזרתה תוכל להעביר קטע מדף תווים מוסיקליים לתכנית בספקטרום.

## אפקטים קוליים

ניתן להפיק מהספקטרום כל מיני אפקטים קוליים, לרוב ע"י הצבת BEEP בתוך לולאה המשנה במהירות את ערך גובה הצליל. נסה תכניות אלה ושנה אותן כדי לפתח צלילים משלך. שים לב לכך שערכי משך הצליל קצרים מאד, ומגיעים עד מאית שנייה. לחץ על BREAK לעצירת תכניות הלולאות.

## בעבוע

```
10 LET P=INT (RND*40) -30
20 BEEP 0.05,P: BEEP 0.05,P+7:
30 BEEP 0.05,P+1
30 GO TO 10
```

תכנית זו מנגנת קבוצה של 3 תווים שוב ושוב בגובה צליל אקראי. טווח גובה הצליל רחב, אך תוכל לשנות את הערכים בשורה 10 כדי לשנותו.

## מכונה

```
10 FOR X=12 TO 36
20 BEEP .01,X
30 BEEP .01,24-X
40 NEXT X
50 GO TO 10
```

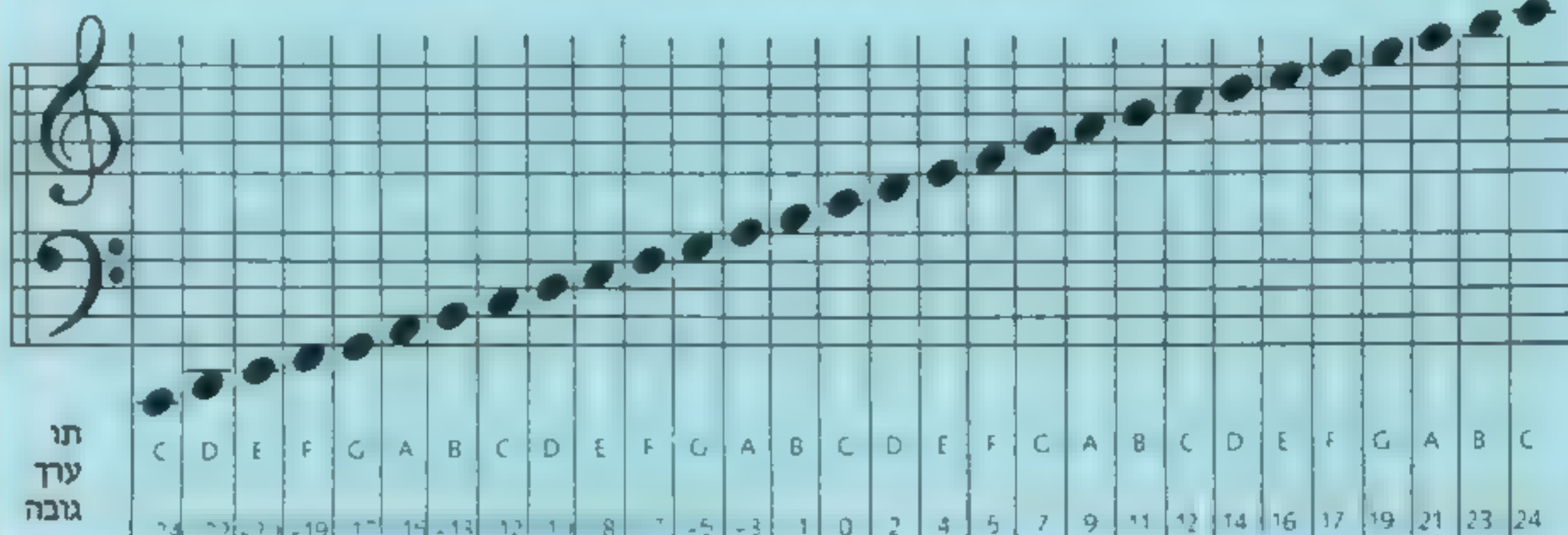
תכנית זו מפיקה שני צלילים, האחד עולה בגובה בעוד השני יורד. זה נובע מהצהרות BEEP המשמיעות שני צלילים שוב ושוב בהפרש של מאית שניה ביניהם בגבהים שונים.

## טבלת ערכי גובה צליל ליצירת מוסיקה

להלן מופיעים ערכי גובה הצליל בספקטרום מהנמוכים ועד הגבוהים.

הוסף 1 לערך גובה הצליל לקבלת צליל חד; הפחת 1 לקבלת צליל

שטוח.





האפקטים הקוליים שהספקטרום מפיק נשמעים היטב על רקע תנועה במסך. כדי להדגים כיצד תוכל להוסיף קול לתכניות, חזור לתכנית העכביש המתרוצץ בעמ' 34.

זכור שהכנסת הצהרת PAUSE בשורה 135 כדי להאיט את התנועה. במקום להשהות את התכנית בדרך זו, תוכל לתכנת השהייה המפיקה צליל. שנה שורה 135 לשורה הבאה:

```
135 GOSUB 500
```

כעת הוסף את השורות הבאות לתכנית.

```
200 STOP
500 FOR P=40-1 TO 30-1 STEP -1
510 BEEP 0.02,P
520 NEXT P
530 RETURN
```

הרץ את התכנית והעכביש יוציא צליל מסולסל בעת צניחתו. השיגרה מנגנת 3 צלילים במהירות רבה הנעשית איטית ככל שהעכביש צונח לנקודה נמוכה יותר במסך. נסה להוסיף תזזית נוספים תוך שינוי שורה 500 והאצה או האטת הצלילים ע"י שינוי הערך 0.02 בשורה 510.

```
10 FOR P=1 TO 40 STEP 0.2
20 BEEP .01,P BEEP .01 P-6
30 NEXT P
```

תכנית זו דומה לתכנית המכונה, אך כעת שני צלילים עולים יחד בהפרש של 6 חציטונים, בנוסף, ערכי גובה הצליל משתנים בצעד של 0.2 – חמישית של חציטון – בכל פעם. זה גורם לצליל לעלות בגובה באיטיות. נסה שינוי גובה צליל קטנים אחרים תוך שינוי ערך STEP.

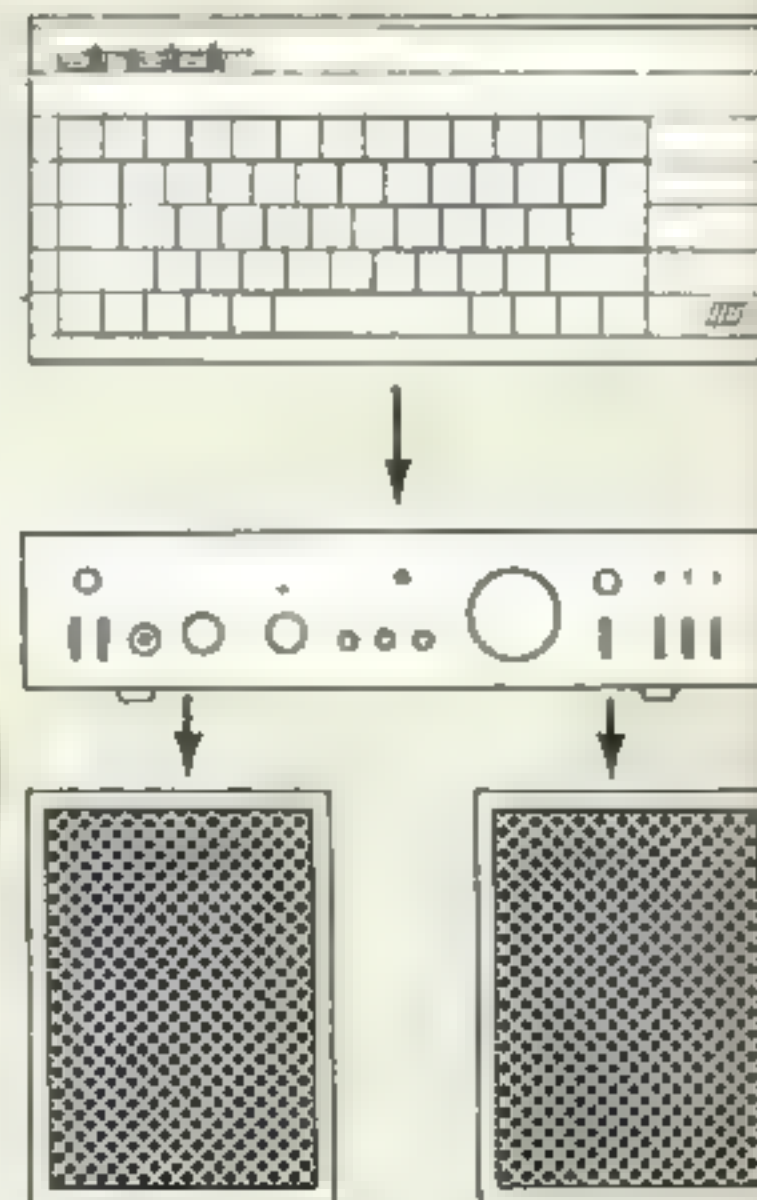
```
10 LET P=CODE INKEY$
15 IF P=0 THEN GO TO 10
20 BEEP .01,(P-30)/2
30 GO TO 10
```

תכנית זו ממחינה שתקיש על מקש כלשהו. כאשר תעשה זאת, כל אחד יפיק צליל אחר. שים לב שלחיצת CAPS SHIFT תוך לחיצה מטה על מקש אחר גורמת לצליל לרדת. תכנית זו פועלת משום ש-CODE INKEY\$ נותן ערך אחר ל"ק כל אימת שמקש חדש נלחץ. השורה השנייה עוצרת את המחשב מהשמעת צלילים אם לא נלחץ כל מקש. ניתן לראות את הערכים ש-CODE מחזיר בטבלת מערך התזזית בעמ' 51.

הספקטרום למערכת שמע גדולה אם תרצה צליל מלא באמת. כאן תצטרך כבל מיוחד עם תקע פין 3.5 מ"מ המתאים לספקטרום ותקע שיכנס לשקע הקלט במגבר או מערכת השמע. הספקטרום מפיק אות קו הדומה לזה המופק מנגני קלטות (Cassette deck) ורשמי סרט, כך ששקע REPLAY או LINE IN במגבר אמור לפעול. אם מתעוררות בעיות כלשהן, התייעץ בחנות לציד שמע.

לשקע MIC של הרשמקול. הוצא החוצה את הקלטת במידת הצורך, הפעל את הרשמקול ואז לחץ PLAY, REWIND (REVERSE) או FAST FORWARD (CUE). וסת את עצמת הקול ברשמקול ותוכל לשמוע את צלילי המחשב בוקעים מהרשמקול ברשמקול. לחילופין, תוכל לחבר אוזניות לרשמקול, אם תרצה. אתה יכול גם לחבר את

כדי להגביר את עצמת הקול בספקטרום, תוכל לחבר את שקע EAR לאוזניות או למגבר ורשמקול. למגבר יהיה בקר עצמת קול בעזרתו תוכל לווסת את עצמת הצלילים. הדרך הפשוטה ביותר לעשות זאת היא להשתמש בכבל הקלטת של הספקטרום לצורך חיבור שקע EAR





# כיצד לשמור את תכניותיך

לא יעבור זמן רב, ותרצה לשמור את תכניותיך על סרט קלטת. כדי לעשות זאת, עליך לחבר רשמקול קסטות לספקטרום ולשמור את התכנית הנמצאת

במחשב. הספקטרום שולח את התכנית לרשמקול הקלטות בצורה בה ניתן להקליטה על סרט. ואז, כאשר תרצה להשתמש בתכנית, תוכל לטעון אותה מהרשמקול למחשב בעזרת תהליך הטעינה המתואר בעמ' 14-15. בפרק שלהלן תוכל לראות כיצד לשמור תכניות ואיך לוודא שהתכנית אכן נשמרה כראוי.

הקלטת תכנית על סרט

**1** ראשית חבר את הספקטרום לרשמקול קלטות מתאים תוך שימוש בכבל הקלטת כמתואר בעמ' 14, אך ודא שרק שקע MIC בספקטרום אכן מחובר לרשמקול.

**2** אם יש בקרי עצמת הקלטת ועצמת קול ברשמקול, ווסת אותם לשני שליש מהעצמה המירבית. אם לא, אל תדאג מכיוון שעצמת ההקלטת תווסת באופן אוטומטי.

**3** הקש פנימה SAVE מלווה בשם התכנית בין מרכאות, לדוגמה:

SAVE "prog 2"

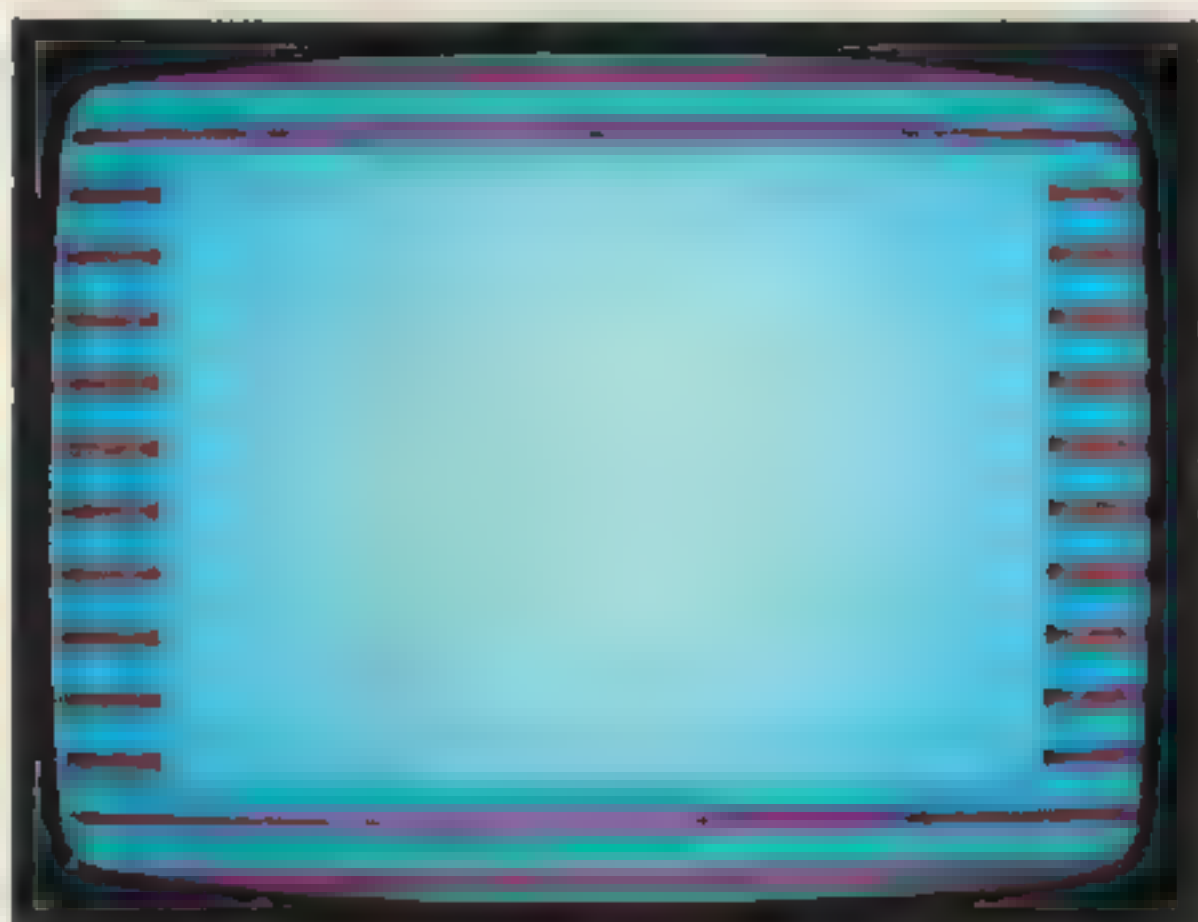


ניתן להשתמש בכל צירוף שהוא של עד 10 אותיות ומספרים. כעת לחץ ENTER הקו של SAVE יעלם ואז תוכל לראות את הוראות הפעלת הקלטת מתוך הספקטרום.



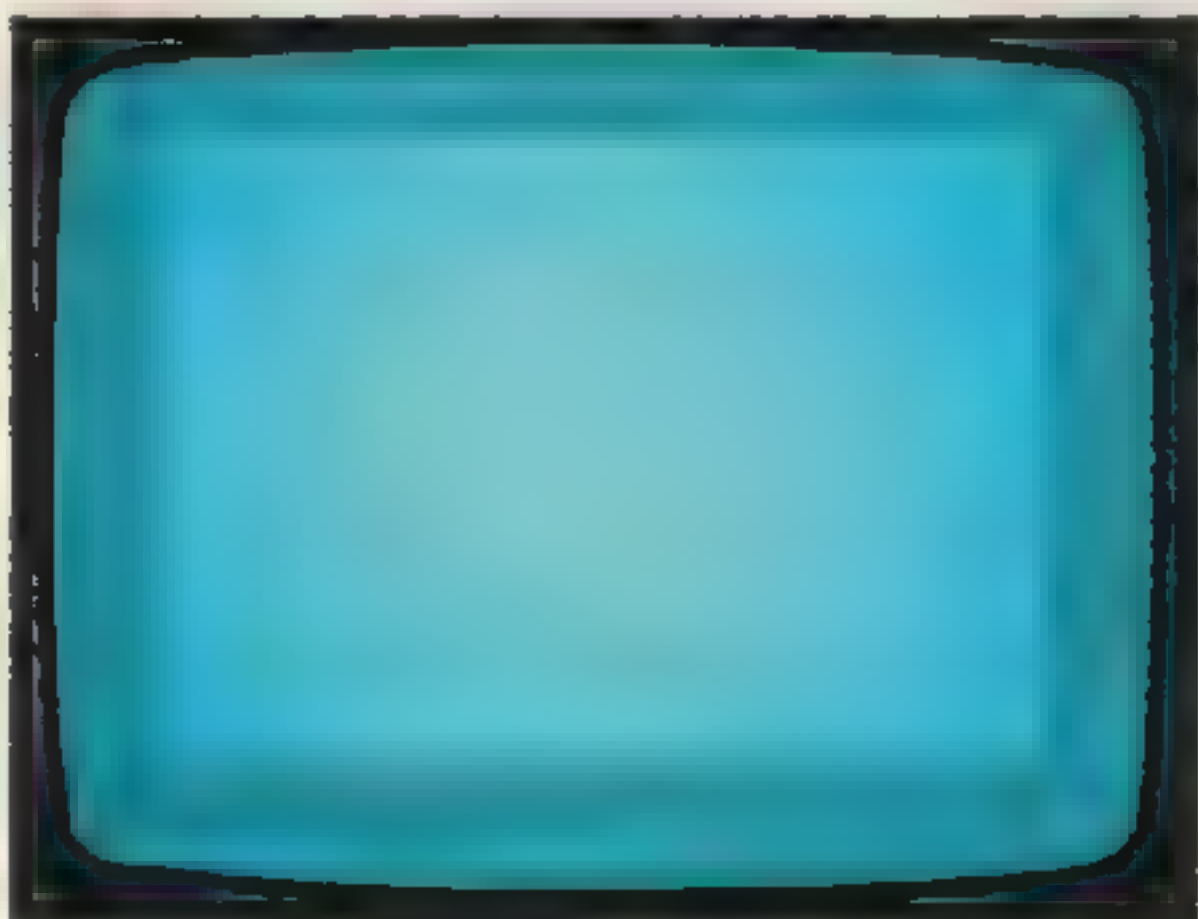
**4** הצב את הרשמקול במצב הקלטת, בעזרת לחיצה על PLAY RECORD יחד. אחר כך לחץ על מקש כלשהו בספקטרום.

**5** כעת חכה עד שהספקטרום יסיים את העברת התכנית. ראשית תראה פסים כחולים ואדומים נעים באיטיות על המסך.



אחר כך יופיע פרץ קצר של פסים כחולים וצהובים. זה קורה בזמן שהספקטרום שולח את שם התכנית לסרט.

**6** על המסך יופיע מרווח קצר ואחריו עוד פסים כחולים ואדומים. זה ילווה בפסים הכחולים והצהובים שנית כאשר הספקטרום שולח את התכנית לסרט. העברת תכנית ארוכה עשויה לקחת זמן רב למדי.



**7** כאשר התכנית נשלחה לסרט, יופיע הדווח 1: OK. עצור את הסרט. התכנית שמורה כעת בסרט. אם תרצה ניתן עתה לבדוק זאת או "לאמת".



1. כתוב את שם התכנית על תווית או כרטיס הקלטת כאשר היא מוקלטת. השתמש באותן אותיות גדולות וקטנות המופיעות על המסך. אם יש מונה ברשמקול העזר בו לאיתור התכנית ורשום את מספר המונה על התווית ליד שם התכנית.
2. לפני שתקליט את התכנית, רשום את שם התכנית בתחילת התכנית בעזרת הצהרת REM, לדוגמה:

5 REM SPIDER Program Version 3

המחשב מתעלם מכל הצהרות REM בזמן ריצת התכנית, וניתן להשתמש ב־REM לצורך הערות ותזכורות בגוף התכנית, לפי הצורך.

אם אינך מקבל דיוח זה, סימן שיש תקלה כלשהי. ראשית בדוק בתרשים הזרימה "איתור תקלות טעינת תוכנה" בעמ' 16, מכיוון שיתכן שהתכנית מוקלטת כראוי על הסרט אך אינה נטענת שנית למחשב לצורך אימות. אם התקלה כאן, תקן אותה, גלגל חזרה את הסרט ואמת את התכנית שנית. אם המחשב עדיין לא מאמת את התכנית, עיין בתרשים הזרימה "איתור תקלות איחסון תוכנה" בעמוד הבא. אל תלחץ NEW, RESET או שתכבה את המחשב, כי אז תמחק התכנית מהזכרון ללא קיום עותק אמין על סרט.

#### הפעלת תכנית אוטומטית

ניתן להקיש SAVE ואחריו את שם התכנית עם LINE 1, לדוגמה:

SAVE "SPIDER" LINE 1

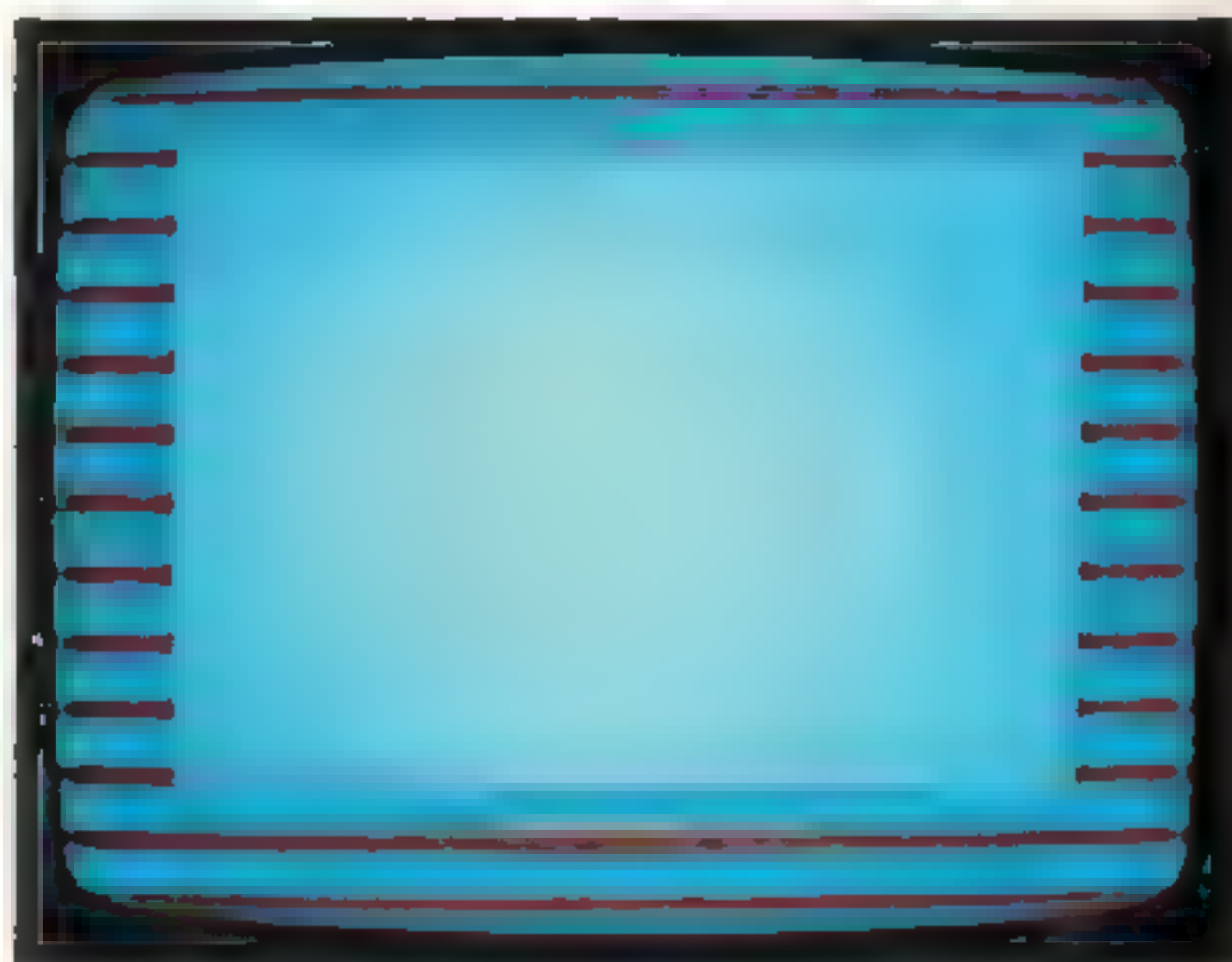
נוהל ההקלטה אינו שונה מקודם אך בזמן האימות, אל תכלול LINE 1 אחרי VERIFY ושם התכנית. תכניות שיוקלטו עם LINE 1 יופעלו אוטומטית בזמן הטעינה. אין צורך להשתמש ב־RUN (אך זכור לעצור את הסרט כאשר התכנית מתחילה). מה שקורה הוא, שהתכנית מתחילה בשורה 1 ואם היא אינה קיימת, המחשב ידלג לשורה הראשונה בתכנית. שינוי 1 למספר אחר יגרום להפעלה אוטומטית של התכנית בשורה המוקשת.

#### שמירה על סרט של \$ SCREEN, CODE ו־DATA

ניתן להשתמש ב־SAVE יחד עם CODE או SCREEN \$ לשמירת קטע מזכרון הספקטרום ועם DATA לאיחסון מערך. ראה את ההסברים ב"מדריך עיין למתכנת".

#### כיצד תאמת את קיום התכנית

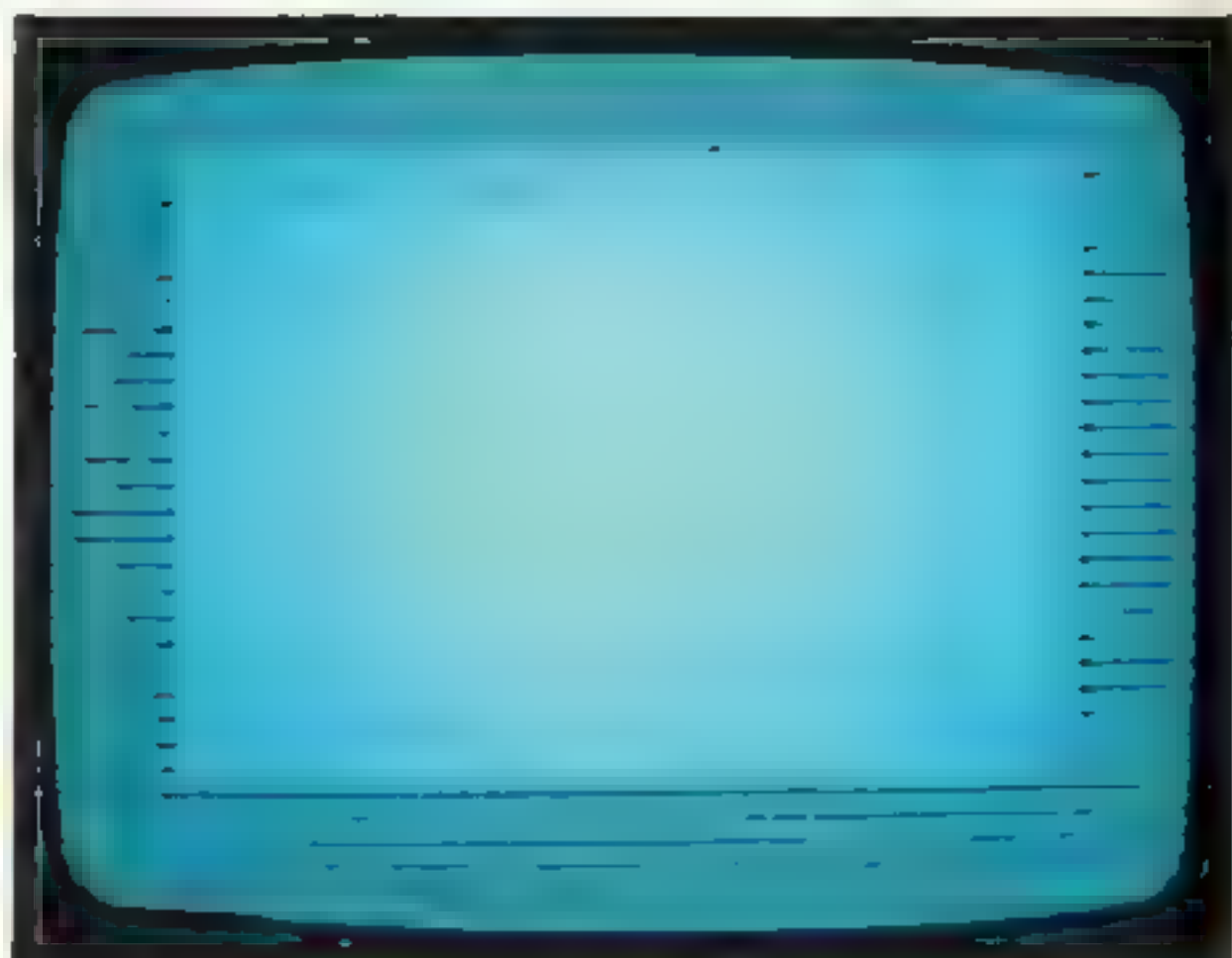
למרות שהמחשב שלח את התכנית לרשמקול הקלטות, אינך יכול להיות בטוח שהתכנית הוקלטה בהצלחה על הסרט. אך הספקטרום מסוגל לבדוק זאת עבורך. נוהל זה נקרא אימות. ראשית הקזר את הסרט לתחילת התכנית, וחבר את שקע EAR בספקטרום לשקע EAR ברשמקול. (ניתן להשאיר את שקעי MIC מחוברים). כעת הקש VERIFY מלווה בשם התכנית במרכאות. לחץ ENTER והפעל את הרשמקול. על המסך יראה אותו רצף של פסים כחולים ואדומים, ורצועות כחולות וצהובות. שם התכנית יופיע וישאר עד סיום תהליך האימות.



כאשר הקטע השני של כחול וצהוב יופיע כמתואר למטה, יראה הדיוח:

0 OK,0:1

זה אומר שהספקטרום השווה את התכנית שעל הסרט עם התכנית שעדיין נמצאת בזכרון ומצא שהן זהות. התכנית אומתה בתוצאה חיובית.



אתה יכול להיות בטוח כעת שהתכנית נמצאת על הסרט.







# למד על הספקטרום + ZX

פרק זה לוקח אותך על תוך קרבי הספקטרום + ZX בהסבירו כיצד פועלים הרכיבים השונים מתחת למקלדת וכיצד הם מתחברים יחד כדי שהמחשב יתפקד. כמו כן מוסבר בפרק כיצד תוכל להשתמש ברכיבים "הקפיים" – התקנים נוספים המאפשרים לשפר את ביצועי הספקטרום ולהביאו לדרגת מערכת מחשב מלאה. לבסוף, תמצא כאן יותר על הצד הטכני במחשב – כולל הצורה בה מאורגן הזכרון, יחד עם המפרט הטכני של הספקטרום.



## מה נמצא בפנים?

קרא הלאה במטרה לגלות – ואל תנסה לפתוח את הספקטרום ZX+ כדי למצוא כיצד הוא פועל. בצורה זו רק תפגע בתקפות האחריות ואתה עלול לגרום לנזק רב.

בתוך הקופסה נמצאים שני מחברי כבל שטוח המקשרים את המקלדת לשאר רכיבי הספקטרום. כל יתר הרכיבים הללו מותקנים על כרטיס מעגל מודפס אחד בלבד. הכרטיס נושא רכיבים חשמליים תקינים כגון נגדים וקבלים, אך הפריטים העקריים הם השבבים הזעירים המרובעים בצבע שחור (ג'זקים), אשר מסודרים ביחידות או בקבוצות.

### מה בתוך השבב

החלק התפקודי של שבב זעיר הוא בעצם קטן בהרבה מהמארז הפלסטי הנושא אותו. המארז מתוכנן בעיקר לתמוך בכל החיבורים שהשבב דורש, ומאפשר לו להתקע בשקעים הנמצאים בכרטיס המעגל. השבב עצמו הוא פיסת צורך דקה אשר מכילה אלפים רבים של צמתים חשמליים. כל צומת פועלת כמתג לצורך עצירה, העברה או אחסון של אותות חשמליים המגיעים אליה. למרות שזהו בעצם תהליך פשוט, קיימים צמתים כה רבים הפועלים יחד עד כי ניתן להפיק אותות המאחסנים או מעבדים מידע במהירות עצומה ודיוק רב. הספקטרום ZX+ כולל מספר שבבים שונים, כשכל אחד מתוכנן לבצע משימה מסוימת לצורך ריצת המחשב.

### כיצד מתחברים השבבים?

בסך הכל, הספקטרום הוא מעגל חשמלי מסובך ביותר. אותות קוד הכוללים דחפים חשמליים חולפים ללא הרף בנתיבים שבתוך ובין השבבים והרכיבים כדי שהמחשב יפעל.

כיצד נשמר כל דבר בסדר הנכון כך שהאות הנכון יגיע למקום הנכון בזמן הנכון? בתוך אחד השבבים טמון שעון המחשב. שעון זה פועל על פליטת דחפים חשמליים – 3.5 מיליון דחפים בשנייה. דחפים אלה נעים בצורה סדירה דרך המעגלים ליצירת אותות קוד אשר מבקרים את פעולות החלקים ושומרים על הסדר הנכון.

קרבי הספקטרום ZX+ במבט זה של כרטיס מעגל הספקטרום ZX+ הוסרו שני מחברי כבל שטוח של המקלדת. כאשר הספקטרום נמצא בשימוש, כל לחיצה על מקש מביאה למגע בין צמד חוטים מתחת למקלדת. מגע זה שולח אות קוד ליעד.

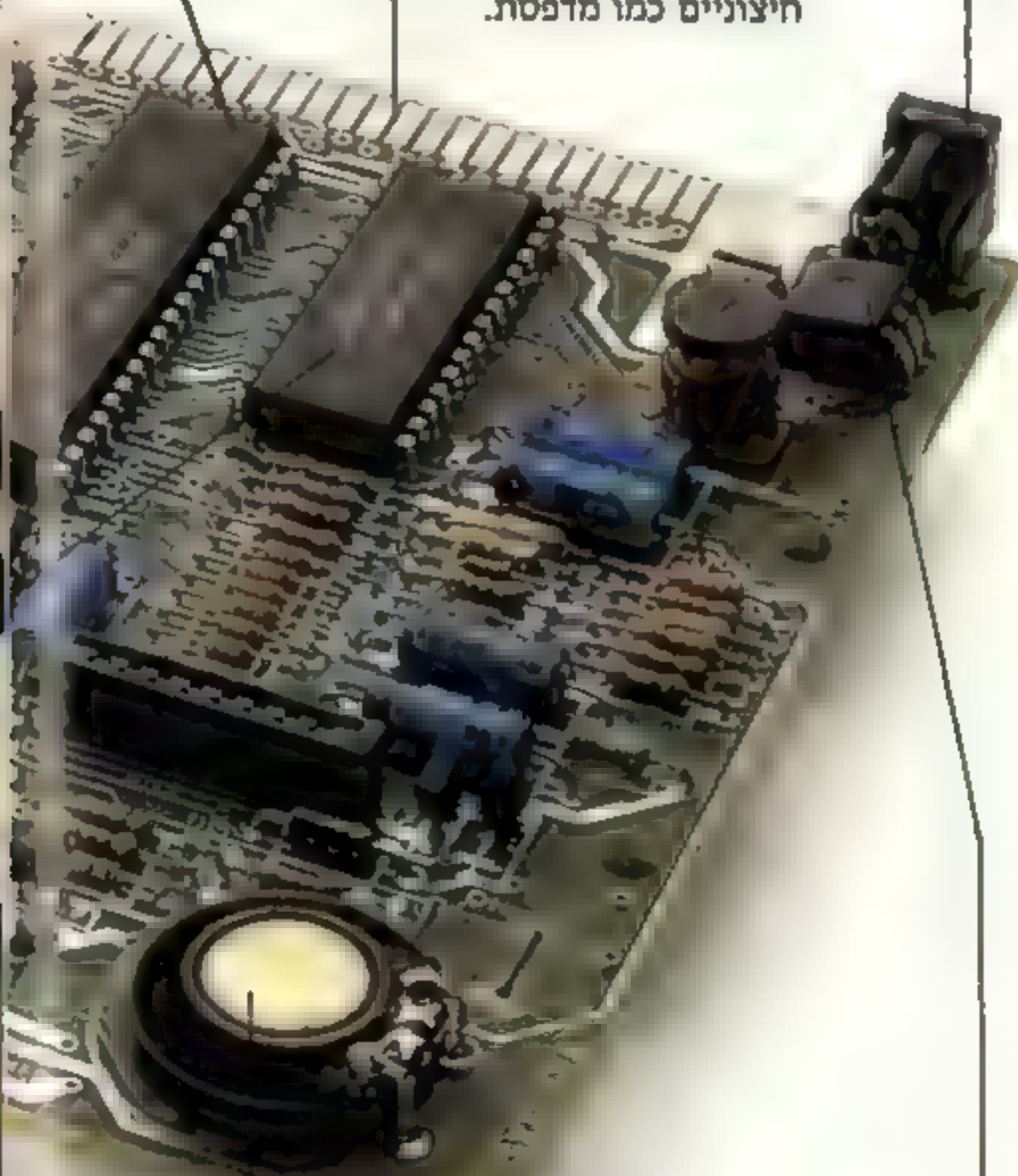
יחידת עיבוד מרכזית (CPU) – ה"מוח" של המחשב, היע"מ הוא מעבד זעיר מדגם 286. המעבד מבצע את כל פעולות החישוב ומבקר את הפעולה הכללית של הספקטרום.

שקע 9VDC – מתחבר לספק הכח.

מחבר קצה – מחבר את הספקטרום להתקנים חיצוניים כמו מדפסת.

וסת מתח – רכיב זה מונע הפרעה למחשב כתוצאה משינויים כלשהם במתח האספקה.

רמקול – מפיק צלילים במידת הצורך.





שקעי קלטת – אלה משמשים להעברת נתונים ותכניות מהזכרון לסרט, ולהזנתם חזרה מסרט לזכרון המחשב.

שבבים לוגיים – שבבים אלה פועלים כמימשק בעת החלפת מידע בין היע"מ והזכרון בגישה אקראית (ראם).

תפוקת טלביזיה – רכיב זה מפיק את האותות הנשלחים למקלט הטלביזיה.

מערך לוגי בלתי תלוי (ULA) – שבב זה מפיק את התצוגה מתוך נתונים הנשמרים בזכרון ראם וכמו כן פועל כבקר של המערכת.

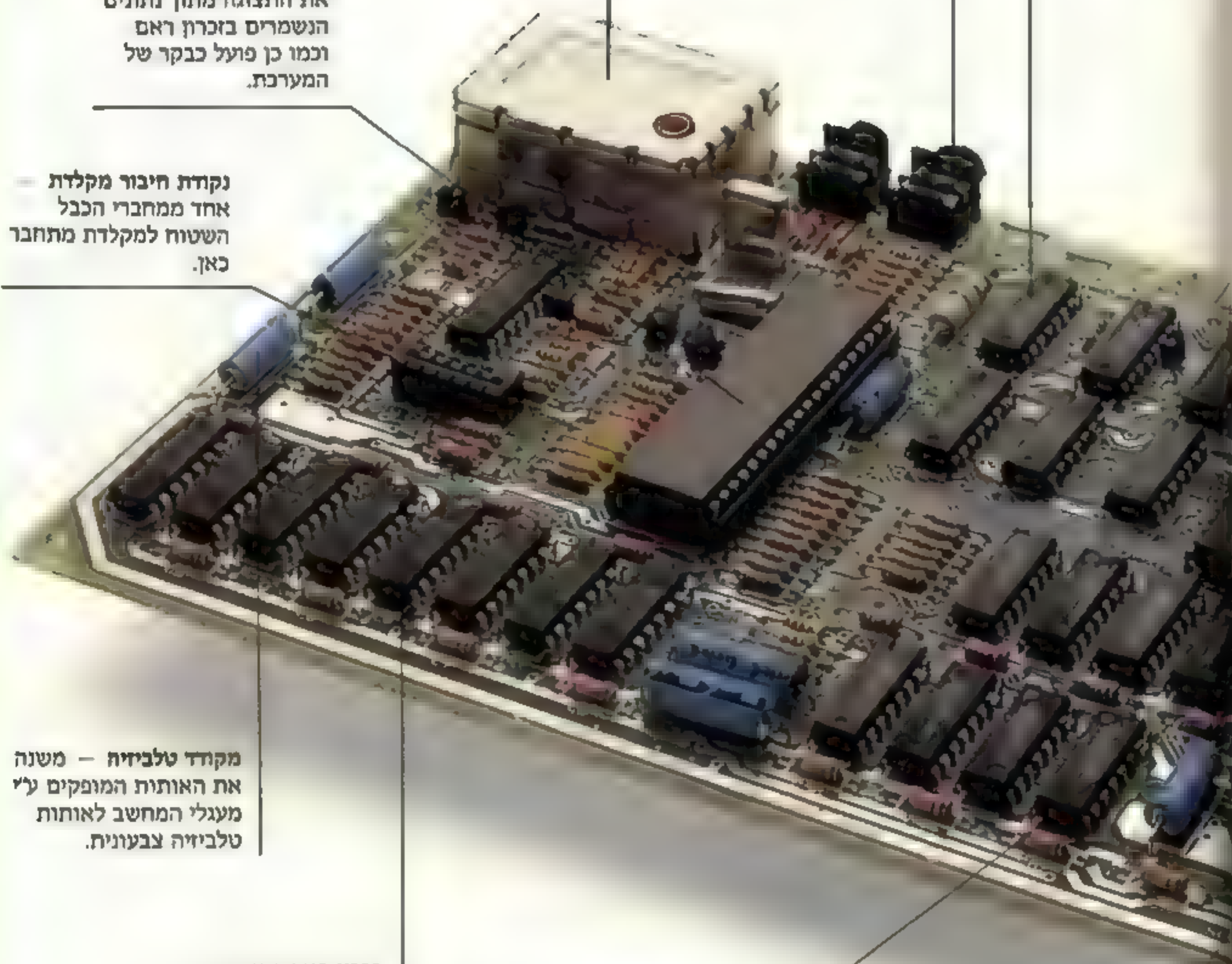
נקודת חיבור מקלדת – אחד ממחברי הכבל השטוח למקלדת מתחבר כאן.

מקדח טלביזיה – משנה את האותות המופקים ע"י מעגלי המחשב לאותות טלביזיה צבעונית.

זכרון בגישה אקראית (RAM) – שבבים אלה מכילים את התכנית המוזנת למחשב וכל נתון מסויים הנדרש ע"י התכנית, כגון ערכים המיוצגים ע"י משתנים. תכולת 48 ק' של זכרון ראם ניתנת לשינוי מן המקלדת, וניתן גם למחקה כליל ע"י כבוי המחשב או לחיצת RESET

זכרון קריאה בלבד (ROM) קטע מהזכרון בן 16 ק' המכיל את הוראות ההפעלה הקבועות הנדרשות ע"י היע"מ. בין היתר, הוראות אלה ממירות את תכניות הבייסיק לשפה שהיע"מ מבין. תכולת שבבי זכרון אלה לא ניתנת לשינוי מן המקלדת.

נקודת חיבור מקלדת – אחד ממחברי הכבל השטוח למקלדת מתחבר כאן.





הספקטרום ZX+, בדומה למערכות מיקרו מחשב אחרות בנוי מארבעה חלקים עקריים. אלה הם יחידות הקלט, כגון מקלדת, אשר מזינה נתונים או תכנית אל המחשב; הזכרון הארעי והזכרון הקבוע, אשר אוגרים נתונים, תכניות והוראות הפעלה; יחידת העיבוד המרכזית CPU, אשר מפעילה את פקודות התכנית על הנתונים, ויחידות הפלט, אשר נותנות את התוצאה.

## הכנסה והוצאת תכנית

10 PRINT 6 + 2

ראשית, אתה מפעיל את המקלדת. מתחת למקשים נמצאת רשת חוטים צולבת. בכל פעם שנלחץ מקש, נוצר מגע בין זוג חוטים השולח אות קוד אל היע"מ. היע"מ בתורו שולח את הקוד לזכרון ראם, בו הוא מאוחסן.

אם תדרוש מהספקטרום לשמור את התכנית על סרט, היע'מ יפנה שוב לזכרון ראם לקבלת הקודים. אך במקום לפעול עליהם, הוא שולח את הקודים ליחידת המרה אשר משנה אותם לאותות קול. אותות אלה נשלחים לרשמקול ומוקלטים על סרט. כאשר תטען מאוחר יותר את התכנית, אותות הקול מהרשמקול ישתנו חזרה לקודי מחשב ע'י יחידת ההמרה. היע'מ ישלח אותם חזרה לזכרון ראם, שם הם יאוחסנו עד שיידרשו.

מספר  
עשיוני 6

מספר  
כינארי

כל הקודים הגורמים לספקטרום לפעול הם בצורה בינארית. הם נקראים בינאריים מפני שכלם מורכבים משני סוגי אותות בלבד. הם ניתנים לייצוג כמספרים בינאריים, כלומר, מספרים המכילים רק שתי ספרות – 0 ו-1. המספר הבינארי עבור 6, למשל, הוא 00000110.

בתוך הספקטרום, הקודים בנויים מרצף של דחפים חשמליים מהירים. אם מגיע דחף חשמלי לנקודה מסוימת, הוא מייצג 1 בינארי. אם דחף לא מגיע תוך פרק זמן מסוים, זה מייצג 0 בינארי. בקוד מחשב, 6 הוא לא-כן-לא-לא-לא-לא-לא. הספקטרום מטפל ב-8 קודים בו זמנית.

בתרשים זה תוכל לראות כיצד המחשב משתמש בסוגים שונים של קידוד בינארי להעברת נתונים ממקום אחד לשני.

אות קוד  
ביצארי האע

קוד בינאי  
השם  
בתאי הזכרת

אות קוד  
בינאר  
הנשלח או  
נשלף  
מהסרט

קוד בינארי  
המוקלט על  
סרט

יש דחף      אין דחף

יש מטען      אין מטען

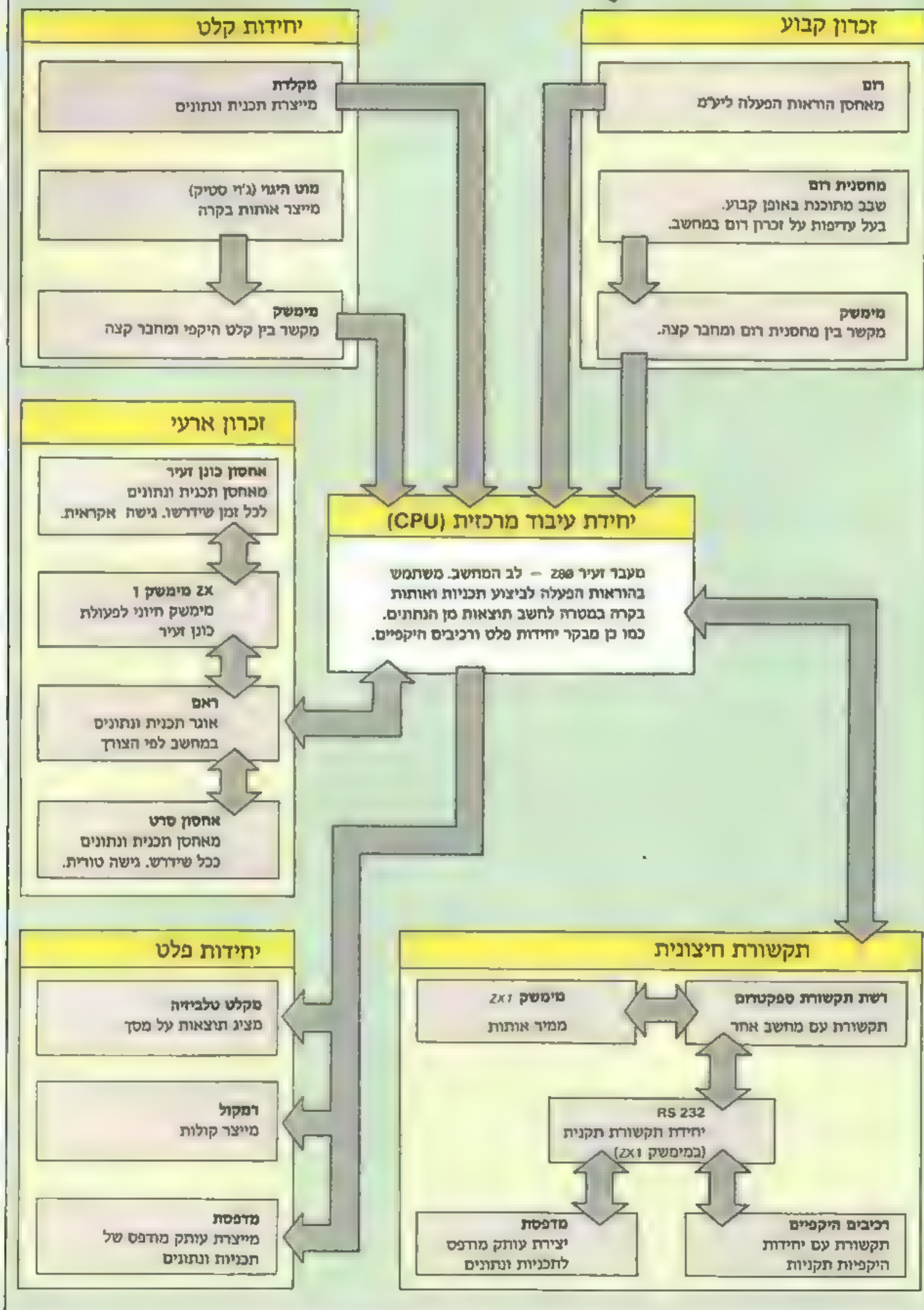
צִלִּיל עֲבוּה      צִלִּיל נִמְח

מגנט      לא מגנט



תרשים זה מתאר כיצד עובר מידע מקחד מיחידות קלט כמו המקלדת, דרך מערכת העיבוד של הספקטרום, ומשם ליחידות פלט כמו מסך הטלביזיה.

חיצים חדכווניים מציינים נתיבים הפועלים בכוון אחד בלבד. חיצים דרכוויים מציינים נתיבים הפועלים בשני הכוונים.





# כיצד לחבר רכיבים היקפיים

## מדפסות תואמות-ספקטרום

מדפסות מסויימות מתחברות ישירות למחבר הקצה בספקטרום. אם יש כבר ברשותך מדפסת סינקלייר ZX, למשל, תוכל לחברה למחשב ללא שימוש במימשק. סוג זה של מדפסת מתחבר גם לגב המימשק ZX1. ברם, כדי להשתמש במדפסות הדורשות פלט RS232, עליך להתחבר לשקע D במימשק ZX1.

ניתן לשפר את ביצועי הספקטרום ZX+ ולהגיע למערכת מחשב מושלמת ורבת עצמה תוך שימוש ברכיבים היקפיים של סינקלייר ותואמי ספקטרום אחרים. הרכיב המרכזי במערכת הוא מימשק ZX1, המאפשר לך לחבר כוננים זעירים לטיפול מהיר ונוח בתכניות ונתונים, ומתחבר גם למגוון רחב של רכיבים היקפיים שונים, כולל מחשבי ספקטרום אחרים. בעזרת מימשק זה, תוכל לחבר את הספקטרום למדפסות תקניות, ולחבר את המחשב למודם המאפשר שידור וקליטת תכניות ונתונים דרך קו טלפון. קיימים גם מימשקים אחרים המקשרים מחסניות רום נתקעות למחשב לצורך טעינת תכניות מיידית. בעזרת אלה, תוכל גם לחבר מוט היגוי כדי להשיג בקרה נוחה על משחקי מחשב.

מימשק תקליטון רום/מוט היגוי

מחבר קצה – רכיבים היקפיים מתחברים למחשב דרך מחבר זה.

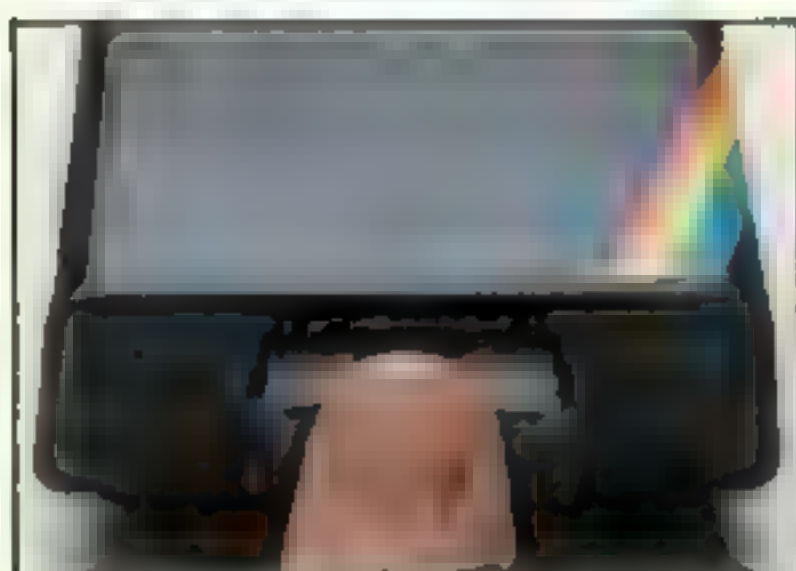
אזהרה – רכיבים היקפיים חייבים להיות מחוברים לפני הכנסת מתח למחשב.





#### מחסניות רום ומוטות היגוי

מימשקים כמו מימשק ZX2 מאפשרים חיבור מחסניות רום ומוטות היגוי. מחסניות רום נטענות מייד עם ההפעלה כדי לטעון במהירות תכניות בעלות זמן טעינה גדול מסרט.



טעינת כונן זעיר – תקליטוני כונן זעיר מוכנסים לחריץ בחזית הכונן.



חיבור מחסנית רום – המחסנית מונכסת לשקע המימשק. כאשר המחשב מופעל, התכנית נסענת אוטומטית, ועוקפת את זכרון רום במחשב.

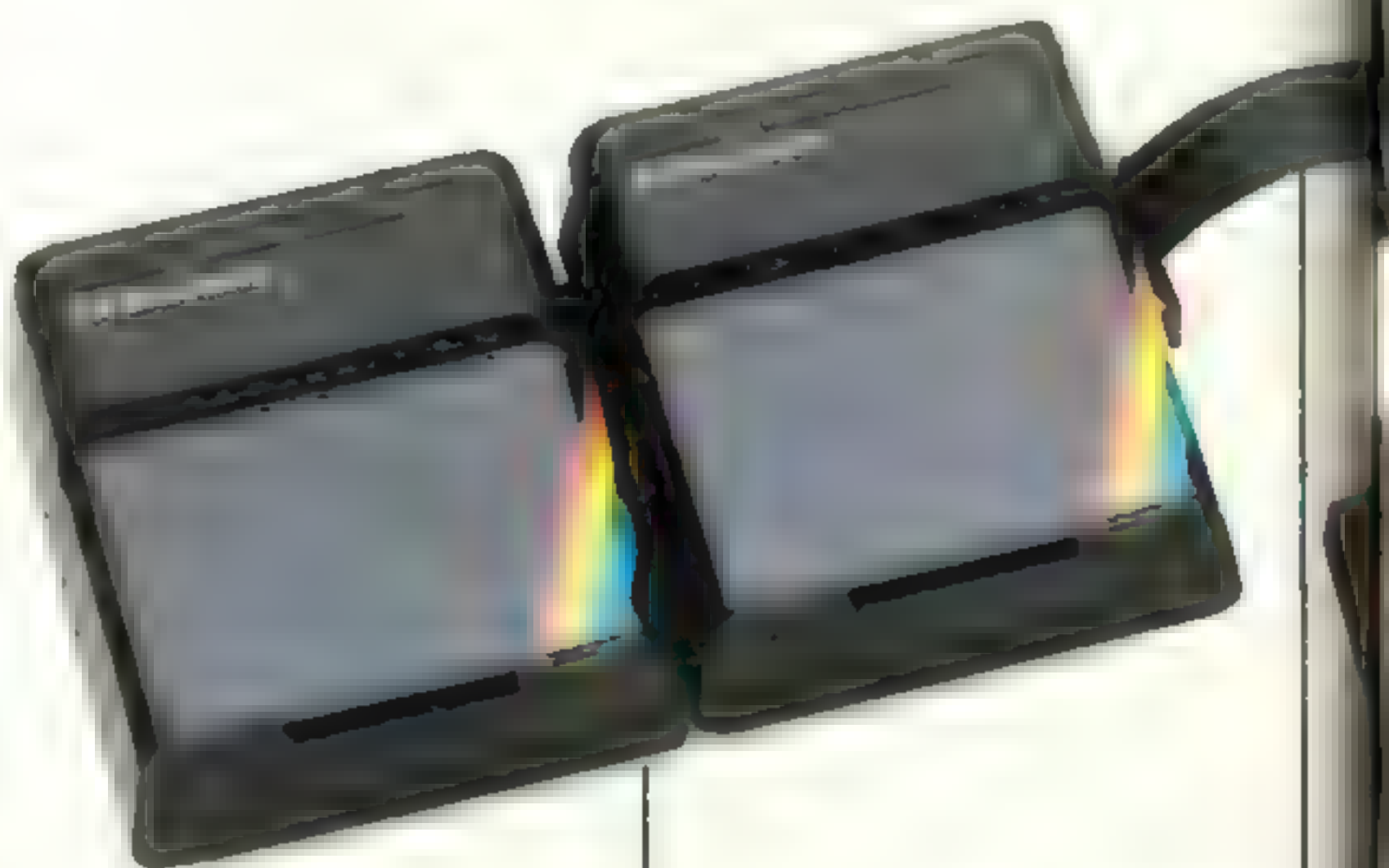
#### מימשק ZX1

יחידת מימשק ZX1 מתחברת לבסיס ולחלק האחורי של הספקטרום. היחידה מקשרת את הספקטרום לעד 8 כוננים זעירים, עד 63 מחשבי ספקטרום אחרים, ובעזרת יחידת המימשק התקני RS232, למגוון רחב של רכיבים היקפיים תקינים.

כוננים זעירים ותקליטונים מחליפים את הרשמקול והסרטים לצורך אחסון תכניות ונתונים. על ידי הכנסת תקליטוני כונן זעיר תוכל לשמור, לאמת ולטעון תכניות תוך שניות. כל תקליטון יכול לאחסן עד 85 ק' נתונים ובעזרת השימוש במקסימום של 8 כוננים זעירים, יהיה לספקטרום קיבול מאגר מקוון (On-line) של עד 680 ק'! כל תכנית מאותרת באופן אוטומטי עם זמן גישה טיפוסי של 3.5 שניות. בעזרת כבל רשת התקשורת המסופק עם יחידת המימשק, תוכל לקשר את המחשב עם מחשב אחר – ספקטרום ZX או מחשב ספקטרום ZX+ נוסף. רשת תקשורת זו ניתנת להרחבה עד 63 ספקטרומים אחרים. הנתונים מועברים ביניהם בקצב של 10000 תוים בשנייה.

יחידת מימשק ZX1 כוללת גם מימשק RS232 עם שקע S בן 9 מגעים אשר דרכו ניתן לחבר לספקטרום מדפסות, רכיבים היקפיים תקינים אחרים, מודמים ומחשבים אחרים. כמו כן קיים גם כבל מימשק תקני.

חיבור רכיבים היקפיים – בעת הפעולה, מימשק ZX1 נתקע במחבר הקצה כך שהוא נמצא מאחורי ומתחת למחשב. האיור כאן מראה את המערכת לפני חיבור המחשב למימשק.



כבל שטוח – הכבל השטוח מחבר את הכונן הזעיר למחשב דרך יחידת מימשק ZX1

יחידות כונן זעיר – ניתן לחבר עד 8 יחידות איחסון כאלה לספקטרום אחד.

הערה – לספקטרום ZX+ יש שתי רגליים פנימיות לצורך הטיית המקלדת. אין להשתמש ברגליים אלה כאשר מורכב מימשק ZX1



# מיפוי זכרון בספקטרום ZX+

אם תתבונן בתצלום של קרבי הספקטרום בעמ' 42-43, תראה שקיים שבב רום אחד, 16 שבבי ראם קטנים יותר. שבבים אלה מהווים את זכרון הספקטרום. הזכרון מורכב מ-65536 יחידות אחסון שכל אחת מהן מכילה בית אחד (מספר בתחום שבין 0 ועד 255). כל יחידה מזוהה על ידי מספר הנקרא כתובת.

רום הוא קיצור מאנגלית עבור זכרון קריאה בלבד. זהו חלק מהזכרון אשר מכיל הוראות הפעלה עבור יחידת העיבוד המרכזית. זהו רום 16 ק', שפירושו הוא זכרון בתכולה של  $16 \times 1024$  (16384) בתים או כתובות. הבתים ניתנים לקריאה בלבד מזכרון זה, כך שאינם ניתנים לשינוי. (אילו היו ניתנים לשינוי, המחשב היה מפסיק לפעול.) אתה יכול לבחון את הבית בכל כתובת בעזרת פקודת PEEK.

ראם פירושו זכרון בגישה אקראית, והוא מכיל את התכניות והנתונים המוחנים למחשב. לספקטרום יש ראם 48 ק', כלומר, הזכרון מכיל  $48 \times 1024$  (49152) בתים או כתובות. גישה אקראית פירושה, שבית בכל כתובת שהיא ניתן לשינוי, והדבר ניתן לביצוע בעזרת הפקודה POKE.

כתובות הזכרון משתרעות מ-0 עד 65535, כשהרבע הראשון עבור זכרון רום והיתר לזכרון ראם.

## משתני מערכת

בעמודה שלהלן מתואר ארגון הזכרון בספקטרום. בתרשים זה תוכל לראות היכן נמצאים החלקים השונים אשר מבקרים את המחשב. כמה מאלה עשויים לשנות מיקום, וגבולותיהם נתונים בתחום של משתני מערכת.

משתני המערכת בספקטרום אינם משתנים כמו אלה הקיימים בבסיסיק. אלה הם פשוט שמות לערכים שימושיים מסויימים הנמצאים בכתובות או מיקומים מסויימים בזכרון. מטרת השם היא לעזור לך לזכור את משמעות הערך המסויים השמור במקום זה. לדוגמה, משתנה המערכת RAMTOP הוא הכתובת העליונה בזכרון ראם. תחום זכרון זה מכיל תכנית בייסיק ואת ערכי משתניה. כתובתו של RAMTOP היא 23730.

## מפת זכרון

ראם 48 ק'

RAMTOP

גרפיקה  
בהגדרת משתמש

מחסנית GOSUB

פנוי

STKEND

מחסנית מחשבון

STKBOT

שטח עבודה זמני

נתוני קלט

WORKSP

פקודה או שורה  
העוברת עריכה

E-LINE

משתנים

VARS

תכנית בייסיק

PROG

נתוני ערוץ

CHANS

מיפוי כונן זעיר

23734

משתני מערכת

23552

מאגר מדפסת

23296

מאפיינים

22528

קובץ תצוגה

16384

רום 16 ק'



# למד על הבייסיק של סינקלייר

פרק זה מתאר את הבייסיק של סינקלייר במלואו. כאן תמצא תקציר של אופן פעולת כל מילת מפתח, ופרטים נוספים על פעולת הבייסיק של סינקלייר. המידע הניתן כאן משתרע ממה שנדרש לצורך תכנית בייסיק הפשוטה ביותר ועד למתקדמת ביותר. פרק זה אינו מיועד לקריאה שוטפת מתחילתו ועד סופו. למעשה, זהו מילון למתכנת אשר מאפשר לך לנצל את מלוא יכולתו של הספקטרום.



# מדריך עיון למילות מפתח עבור מתכנת בייסיק של סינקלייר

מדריך זה מכיל תאורים מפורטים של כל מילות המפתח בבייסיק אשר קיימות בספקטרום ZX+. כל רשומה כוללת

- מיקום מילת המפתח
- סוג מילת המפתח
- מטרת מילת המפתח
- השימוש במילת המפתח
- פורמט תיכנות

הפרטים הניתנים עבור מיקום, מטרה ושימוש מסבירים את עצמם. סוג ופורמט כוללים הגדרות מסובכות יותר, וכדי להפיק את מירב התועלת ממדריך זה, עליך לקרוא תחילה בעיון את המידע בעמוד זה.

## סוגי מילות מפתח

מילות מפתח נופלות בתחום אחד או יותר מארבעת הסוגים הבאים.

### פקודה

מילת מפתח הגורמת לביצוע פעולה וניתנת לשימוש לביצוע פקודה ישירה. פקודה מתבצעת מייד לאחר הקשתה. דוגמאות – LOAD, RUN

### הצהרה

מילת מפתח הגורמת לביצוע פעולה וניתנת לשימוש בשורת תכנית. היא מתבצעת רק כאשר התכנית מורצת. דוגמאות – INPUT, DRAW

### פונקציה

מילת מפתח היוצרת ערך מסוג כלשהו. היא יוצרת חלק מפקודה או הצהרה. דוגמאות – INT, RAND

### אופרטור לוגי

מילת מפתח המשמשת לביטוי לוגיקה בהצהרה או פקודה. האופרטור יכול לקבוע או לשנות את אמיתות תנאים מסויימים. בספקטרום יש שלשה אופרטורים לוגיים – NOT, OR, AND

## מספרים ומשתנים

### מספרים

מאוחסנים בדיוק של עד 9 או 10 ספרות. טווח הטיפול במספרים הוא מ- $10^{36}$  ועד ל- $10^{-4}$ .

### משתנים תקפים

משתנים מספריים בכל אורך, המתחילים באות. רווחים לא נחשבים וכל האותיות מומרות לאותיות קטנות. אין הבחנה בין אותיות גדולות וקטנות.

משתני מחרוזת עם אות בודדת כלשהי, בלחית סימן \$. אין הבחנה בין אותיות גדולות וקטנות.

משתני מערכים, ראה הסבר בנושא DIM, הן גם במשתני מערך ותיות.

## פורמט מילת מפתח

וגורמים אחרים כמו ערכים ומשתנים. הקיצורים הבאים משמשים לצורך הגדרת הפורמט.

פורמט מילת המפתח מתאר את התחביר של כל מילת מפתח – כלומר, הצירוף הנכון של מילת המפתח

קיצור	הסבר	דוגמה
num-const	קבוע מספרי (מספר)	24.5
num-var	משתנה מספרי (משתנה העשוי להכיל מספר)	sum
num-expr	בטוי מספרי (כל צירוף תקף של קבועים מספריים, משתנים ומילות מפתח הנותנים מספר).	sum+24.5 RND*7
int-num-const int-num-var int-num-expr	קבוע, משתנה או בטוי מספרי אשר ערכו מעוגל לשלם הקרוב ביותר.	
string-const	קבוע מחרוזת או מחרוזת (כל צירוף של תוים בין סימני מרכאות)	"ZX Spectrum +"
string-var	משתנה מחרוזת (משתנה העשוי להכיל מחרוזת).	a\$
string-expr	בטוי מחרוזת (כל צירוף תקף של קבועי, משתני ומילות מפתח מחרוזת, הנותנים מחרוזת).	a\$ + "ZX Spectrum +" a\$ (6 TO 8)
letter	כל אות קטנה או גדולה	x
letter	כל אות קטנה או גדולה המלווה בסימן \$	B\$ a\$
cond	תנאי או תנאי תנאי בתוך תנאי	x=10 AND t<10
statement	כל הצהרת בייסיק אשר תקפה כאשר יש שימוש בה עם הצהרה אחרת.	IF t>10 THEN STOP PRINT INK 2;x
	פרט רשות (אופציה) העשוי לחזור על עצמו.	[ ]

הערה: המונחים ערך מספרי וערך מחרוזת מופיעים בטקסט עבור פרטים מספריים או פריטי מחרוזת בהתאמה.



## סימנים בביוסיק של סינקלייר

סימן	מיקום	פעולה/שימוש	סימן	מיקום	פעולה/שימוש
\$	4	משתנה מחרוזת	-	L	שווה ל...
'	7	התחלת שורה חדשה	:	Z	הפרדת הצהרות בשורת תכנית
(	8	פתח סוגריים	/	V	חילוק
)	9	סגור סוגריים	*	B	כפל
<=	Q	קטן מ... או שווה ל...	.	מקש נפרד	נקודה עשרונית
<>	W	לא שווה ל...	.	מקש נפרד	מציג בעמודה הבאה
>=	E	גדול מ... או שווה ל...	.	מקש נפרד	מפריד בין הצהרות בתוך הצהרת תכנות
<	R	קטן מ...	"	מקש נפרד	פתיחת וסגירת מחרוזת
>	T	גדול מ...	.	מקש נפרד	מציג בעמודה 8 או 16
↑	H	העלה בחזקה	.	מקש נפרד	הפרדת ערכים המלווים מילות מפתח
-	J	חיסור / שלילי			
+	K	חיבור / חיובי / שרשרת מחרוזות			

## מערך התווים בספקטרום ZX+

	0	1	2	3	4	5	6	7	8	9
0					TRUE VIDEO	INV VIDEO	PRINT control	EDIT	cursor left	cursor right
10	cursor down	cursor up	DELETE	ENTER	number	GRAPHICS MODE	INK control	PAPER control	FLASH control	BRIGHT control
20	INVERSE control	OVER control	AT control	TAB control						
30			space	!	"	#	\$	%	&	'
40	(	)	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	@	A	B	C	D	E
70	F	G	H	I	J	K	L	M	N	O
80	P	Q	R	S	T	U	V	W	X	Y
90	Z	[	/	]	↑	-	£	a	b	c
100	d	e	f	g	h	i	j	k	l	m
110	n	o	p	q	r	s	t	u	v	w
120	x	y	z	{		}	~	©	□	◻
130	◻	◻	◻	◻	◻	◻	◻	◻	◻	◻
140	◻	◻	◻	◻	GRAPHICS A	GRAPHICS B	GRAPHICS C	GRAPHICS D	GRAPHICS E	GRAPHICS F
150	GRAPHICS G	GRAPHICS H	GRAPHICS I	GRAPHICS J	GRAPHICS K	GRAPHICS L	GRAPHICS M	GRAPHICS N	GRAPHICS O	GRAPHICS P
160	GRAPHICS Q	GRAPHICS R	GRAPHICS S	GRAPHICS T	GRAPHICS U	RND	INKEYS	PI	FN	POINT
170	SCREENS	ATTR	AT	TAB	VALS	CODE	VAL	LEN	SIN	COS
180	TAN	ASN	ACS	ATN	LN	EXP	INT	SQR	SGN	ABS
190	PEEK	IN	USR	STRS	CHRS	NOT	BIN	OR	AND	<=
200	>=	<>	LINE	THEN	TO	STEP	DEF FN	CAT	FORMAT	MOVE
210	ERASE	OPEN #	CLOSE #	MERGE	VERIFY	BEEP	CIRCLE	INK	PAPER	FLASH
220	BRIGHT	INVERSE	OVER	OUT	LPRINT	LLIST	STOP	READ	DATA	RESTORE
230	NEW	BORDER	CON-TINUE	DIM	REM	FOR	GOTO	GO SUB	INPUT	LOAD
240	LIST	LET	PAUSE	NEXT	POKE	PRINT	PLOT	RUN	SAVE	RANDOM-IZE
250	IF	CLS	DRAW	CLEAR	RETURN	COPY				



**ABS ערך מוחלט****מיקום במקלדת**EXTEND MODE  
G**פונקציה**

ABS נותנת את הגודל המוחלט של ערך מספרי, כלומר, הערך ללא סימן חיובי או שלילי.

כיצד להשתמש ב־ABS  
ABS מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה:

50 LET x = ABS(y - z)

ABS מחזירה את הערך המוחלט של ערך מספרי.

דוגמה  
הפקודה

PRINT ABS — 34.2

מציגה 34.2.

**פורמט**ABS num-const  
ABS num-var  
ABS (num-expr)**ACS ארק קוסינוס****מיקום במקלדת**EXTEND MODE  
SYMBOL SHIFT W**פונקציה**

ACS מחשבת ערך זווית מתוך הקוסינוס שלה.

כיצד להשתמש ב־ACS  
ACS מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה:

60 LET x = ACS(y \* z)

הערך המלווה את ACS (y \* z) שלעיל הוא הקוסינוס של הזווית הנדרשת, הנמצא בתחום 1 - עד 1. ACS מחזירה את ערך הזווית ברדיאנים. כדי להמיר רדיאנים למעלות, הכפל את הערך המוחזר ע"י ACS ב־180/PI.

דוגמה  
הפקודה

PRINT 180/PI \* ACS 0.5

מציגה 60, הזווית במעלות שיש לה קוסינוס של 0.5.

**פורמט**ACS num-const  
ACS num-var  
ACS (num-expr)**AND****מיקום במקלדת**

SYMBOL SHIFT Y

**אופרטור לוגי/פונקציה**

AND פועלת כאופרטור לוגי לבדיקת אמיתות הצירוף של תנאים. רק אם כל התנאים הם אמת סה"כ הצירוף הוא אמת. AND משמשת גם כפונקציה לביצוע פעולות בינאריות על שני ערכי מספרים או מחרוזות.

כיצד להשתמש ב־AND  
בתור אופרטור לוגי, AND מקשרת שני תנאים בהצהרה בה יש לבחון את קיום כל התנאים, לדוגמה

```
90 IF x = y + z AND time < 10
  THEN PRINT "Correct"
```

רק אם שני התנאים (x = y + z) ו־time < 10 הם אמת, המחשב יציג Correct. אם אחד או שני התנאים שקרי, אזי כל הצירוף שקרי, ובמקרה זה התכנית תמשיך לשורה הבאה.

AND כפונקציה  
בתור פונקציה, AND יכולה לפעול על שני ערכים מספריים, לדוגמה

50 LET x = y AND z

AND מחזירה את הערך הראשון (y) אם השני (z) לא שווה ל־0, ומחזירה 0 אם הערך השני (z) הוא 0. AND יכולה לפעול גם על משתנה מחרוזת בתנאי שהוא בא לפני AND. אחרי AND חייב תמיד לבוא ערך מספרי, לדוגמה

50 LET a\$ = b\$ AND z

AND מחזירה את הערך הראשון (b\$) אם השני (z) אינו אפס, ומחרוזת ריקה ("") אם הערך השני (z) הוא 0. שים לב לכך שהספקטרום ZX+ מקצה ערך של 1 לתנאי אמת רש לתנאי שקרי, ומכיר כל ערך שאינו אפס כאמת רש כשקר. הוא לא מעריך צירופים של ערכים מספריים בהתאם לסבלות אמת תקינות.

**דוגמאות**

```
60 LET correct = (x = y + z) AND
  time < 10
70 LET score = score + 10 * (1
  AND correct)
80 LET a$ = ("Out Of Time Or Not"
  AND NOT correct) + "Correct"
```

אם שני התנאים בשורה 60 הם אמת, אזי המשתנה המספרי correct מקבל ערך של 1. ואז score מוגדל ב־10 רש a\$ הופך "Correct". אם אחד משני התנאים שקרי, אזי correct מקבל ערך של 0; Score לא משתנה רש a\$ הופך ל־"Out Of Time Or Not Correct".

**פורמט**Cond AND cond  
num-expr AND num-expr  
string-expr AND num-expr**ASN ארק סינוס****מיקום במקלדת**EXTEND MODE  
SYMBOL SHIFT Q**פונקציה**

ASN מחשבת את ערך הזווית מתוך הסינוס.

כיצד להשתמש ב־ASN  
ASN מלווה בערך מספרי. בטוי חייב להיות בין סוגריים, לדוגמה

60 LET x = ASN(y \* z)

הערך המלווה את ASN (y \* z) שלעיל הוא סינוס הזווית הנדרשת ונמצא בתחום שבין 1 - ועד 1. ASN מחזירה את ערך הזווית ברדיאנים. כדי להפוך רדיאנים למעלות, הכפל את הערך המוחזר ע"י ASN ב־180/PI.

דוגמה  
הפקודה

PRINT 180/PI \* ASN 0.5

מציגה 30, ערך הזווית במעלות אשר הסינוס שלה הוא 0.5.

**פורמט**ASN num-const  
ASN num-var  
ASN (num-expr)**AT****מיקום במקלדת**SYMBOL SHIFT I  
ראה: INPUT ;LPRINT ;PRINT**ATN ארק טנגנס****מיקום במקלדת**EXTEND MODE  
SYMBOL SHIFT E



BIN הופכת מספר בינארי למספר עשרוני.

כיצד להשתמש ב-BIN  
BIN מלווה במספר בינארי המורכב מעד 16 תווי 1 ותווי 0, לדוגמה

50 POKE USR "a", BIN 10101010

BIN מחזירה את הערך העשרוני של המספר הבינארי, בד"כ משתמשים בו בשילוב עם POKE ו-USR כלעיל ליצירת תוים גרפיים בהגדרת משתמש, כאשר 1 מייצג פיקסל של צבע דיו רם פיקסל של צבע הנייר.

דוגמה  
הפקודה

PRINT BIN 11111110

מציגה 254, הערך העשרוני של המספר הבינארי.

פורמט

BIN [1][0]

## BORDER

מיקום במקלות

B

הצהרה/פקודה

BORDER מגדירה את צבע גבולות המסך סביב שטח התצוגה.

כיצד להשתמש ב-BORDER  
BORDER ניתנת לשימוש כפקודה ישירה או כהצהרה בתכנית. BORDER מלווה בערך מספרי, לדוגמה

30 BORDER RND \* 7

הערך המלווה את BORDER מעוגל לשלם הקרוב ביותר ומגדיר את צבע גבולות המסך כלהלן:

- 0 שחור
- 1 כחול
- 2 אדום
- 3 סגול
- 4 ירוק
- 5 תכלת
- 6 צהוב
- 7 לבן

שים לב לכך ש-BORDER קובעת גם את צבע הנייר בתחתית המסך. שלא כמו INK ו-PAPER, לא ניתן להכניס הצהרת BORDER בתוך הצהרת PRINT.

פורמט

BORDER int-num-expr

דוגמה

אם תו נמצא בנקודה 16, 11 ומוצג בצבע דיו 3 (סגול), צבע נייר 6 (צהוב) ובהיר אך לא מהבהב, אזי הפקודה

PRINT ATTR (11,16)

מציגה 115 (3 + 8 \* 6 + 64 + 0).

ATTR בצורה בינארית

ATTR מחזירה בית אחד שבו סיבית 7 (המשמעותית ביותר) היא 1 עבור הבהוב או 0 במצב רגיל, סיבית 6 היא 1 עבור בהירות או 0 במצב רגיל, סיביות 3 עד 5 הן צבע הנייר (בשיטה בינארית) וסיביות 2 עד 0 הן צבע הדיו.

פורמט

ATTR (num-expr, num-expr)

## BEEP

מיקום במקלות

EXTEND MODE  
SYMBOL SHIFT Z

הצהרה/פקודה

BEEP גורמת לרמקול להפיק צליל יחיד באורך וגובה נתונים.

כיצד להשתמש ב-BEEP  
ניתן להשתמש ב-BEEP ליצירת הצהרה בתכנית או פקודה ישירה. BEEP מלווה בשני ערכים מספריים, המופרדים בפסיק, לדוגמה

80 BEEP x,y

הערך הראשון (x) נמצא בתחום 0 עד 10 ומגדיר את משך זמן הצליל בשניות.

הערך השני (y) נמצא בתחום 60- עד 69 ומגדיר את גובה הצליל בחצי טונים. מתחת לזו אמצעי אם הוא שלילי או מעל זו אמצעי אם הוא חיובי.

דוגמה  
הפקודה

BEEP 0.5,1

גורמת לצליל דו דיאז שמעל דו אמצעי להישמע במשך חצי שנייה.

פורמט

BEEP num-expr, num-expr

## BIN מספר בינארי

מיקום במקלות

EXTEND MODE  
B

פונקציה

ATN מחשבת את ערך הזווית מתוך הטנגנס שלה.

כיצד להשתמש ב-ATN

ATN מלווה בערך מספרי. בטוי חייב להיות בתוך סוגריים, לדוגמה

60 LET x = ATN (y \* z)

הערך המלווה את ATN (y \* z) במקרה זה) הוא הטנגנס של הזווית הנדרשת. ATN מחזירה את ערך הזווית ברדיאנים. כדי להפוך רדיאנים למעלות, הכפל את הערך המוחזר ע"י ATN ב-180/PI.

דוגמה  
הפקודה

PRINT 180/PI \* ATN 1

מציגה 45, הזווית במעלות אשר הטנגנס שלה הוא 1.

פורמט

ATN num-const

ATN num-var

ATN (num-expr)

## ATTR מאפיינים

מיקום במקלות

EXTEND MODE  
SYMBOL SHIFT

פונקציה

ATTR נותנת את המאפיינים של מיקום תו מוגדר על המסך. אלה הם צבעי הדיו והנייר, מצב בהירות והבהוב של התו במיקום המסוים.

כיצד להשתמש ב-ATTR

ATTR מלווה בשני ערכים מספריים המופרדים בפסיק וסגורים במרכאות, לדוגמה

150 IF ATTR (v,h) = 115 THEN  
GOSUB 2000

הערך הראשון שאחרי ATTR (v) שלעיל) נמצא בתחום שבין 0 עד 23 ומייצג את מספר השורה של המיקום על המסך.

הערך השני (h שלעיל) נמצא בתחום שבין 0 עד 31 ומייצג את מספר העמודה של המיקום. ATTR מחזירה מספר הנע בין 0 ל-255. מספר זה הוא סכום המאפיינים במיקום המוגדר, ומורכב מהגורמים הבאים: צבע דיו - קוד צבע (0 עד 7) צבע נייר - 8 פעמים קוד צבע בהירות - 64 הבהוב - 128



**BRIGHT**

מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT B

הצהרה/פקודה

BRIGHT גורמת לתוים להיות מוצגים בצבעים בהירים יותר מהרגיל.

**כיצד להשתמש ב־BRIGHT**  
BRIGHT ניתנת לשימוש כפקודה ישירה אך משמשת בד"כ ליצירת הצהרה בתכנית. BRIGHT מלווה בערך מספרי, לדוגמה

**80 BRIGHT 1**

הערך המלווה את BRIGHT מעוגל לשלם הקרוב ביותר במידת הצורך, ולכן יהיה 0, 1 או 8. ערך של 1 גורם לכל התוים הבאים אחריו ומוצגים ע"י PRINT או INPUT להופיע בצבעי דיו ונייר בהירים יותר, וערך של 8 גורם למיקומי תוים בהירים להישאר בהירים ולמיקומי תוים רגילים להישאר רגילים כאשר יודפסו עליהם תוים חדשים.

BRIGHT מלווה ב־8 מבטלת את BRIGHT 1. BRIGHT 8 כן שכל התוים שיוצגו אחריהם יהיו רגילים. ניתן להטמיע (להכניס) את BRIGHT בתוך הצהרות תצוגה הנוצרות ע"י PRINT, PLOT, DRAW, INPUT, CIRCLE BRIGHT מלווה את מילת המפתח אך מקדימה את הנתונים או מאפייני התצוגה; היא מלווה באותם ערכים ובנקודה־פסיק; לדוגמה

**50 PRINT BRIGHT 1;****"WARNING"**

ההשפעה של BRIGHT היא איפוא מקומית ושיימה רק לתוים המוצגים, נקודות שיותו, או קווים שיצוירו ע"י הצהרת התצוגה. שים לב לכך ש־BRIGHT 1 מבהירה את צבע הנייר של כל מיקום התו של 8x8 פיקסלים אם אחד הפיקסלים בנקודה זו מותווה בצבע דיו.

פורמט

BRIGHT int-num-expr [;]

**CAT** קטלוג

פקודת טיפול בקבצים עבור כונן זעיר. ראה מדריך כונן זעיר ומימשק 1.

**CHR \$** מחזרת תוים

מיקום במקלדת

EXTEND MODE  
U

פונקציה

התוים ומילות המפתח הקיימים במקלדת יחד עם תוים גרפיים בהגדרת המשתמש יוצרים את מערך התוים בספקטרום.

ע"י שימוש ב־CHR \$ ומספר קוד, ניתן לקבל כל אחד כמחרוזת. מערך התוים כולל גם כמה קודי בקרה אשר משפיעים על אופן תצוגת התוים. קודים אלה ניתנים להפעלה לקבלת תוים מוצגים בעזרת PRINT לפני CHR \$. ניתן למצוא את מערך התוים השלם עם קודי הבקרה עמ' 51.

**כיצד להשתמש ב־CHR \$**  
CHR \$ מלווה בערך מספרי, לדוגמה

**80 PRINT CHR \$ x**

בטיח חייב להיות בתוך סוגריים. הערך המלווה את CHR \$ (x לעיל) מעוגל לשלם הקרוב ביותר. אם הוא בתחום של 32 עד 255, CHR \$ מחזירה תו של מילת מפתח, תו גרפי בהגדרת משתמש, או מילת מפתח כמחרוזת. הספקטרום משתמש בקוד ASCII עבור ערכים בתחום 32 עד 95 ו־97 עד 126. לדוגמה, אם ל־x מוקצה ערך של 65, ההצהרה שלעיל תציג A.

**קודי בקרה של CHR \$**

ערכים שבין 1 ו־31 מחזירים קודי בקרה או שאינם בשימוש. CHR \$ 6 (PRINT פסיק), 8 (מקשר אחורה) ו־13 (שורה חדשה או ENTER) משפיעים על תצוגות במסך אם יוכלו בהצהרת PRINT. ניתן ללוות CHR \$ בקוד הבקרה ונקודה־פסיק, לדוגמה

**60 PRINT "A"; CHR \$ 6; "B"**

הצהרה זו מציגה

A B  
דרך אחת להשתמש בקודי בקרה של CHR \$ הוא ליצור מחרוזת מורכבת המכילה אותם. ההצהרה

**60 PRINT "A" + CHR \$ 6 + "B"**

נותנת בדיוק אותו אפקט כמו הדוגמה הקודמת.

קודים 16 עד 23 משפיעים על צבע ומיקום וניתן להשתמש בכל אחד מהם במחרוזת מורכבת יחד עם

CHR \$ המלווה בערך קוד צבע בתחום 0 עד 7 עבור CHR \$ 16 (בקרת דיו) ו־17 CHR \$ (בקרת נייר), או ע"י 0 או 1 עבור CHR \$ 18 עד CHR \$ 21 (בקר OVER, INVERSE, BRIGHT, FLASH הפקודה)

**PRINT CHR \$ 16 + CHR \$ 3 +  
CHR \$ 17 + CHR \$ 6 + CHR \$  
18 + CHR \$ 1 + "ZX  
SPECTRUM + "**

מציגה ZX SPECTRUM בצבעי אדום וצהוב מהבהבים.

לחלופין, כמו לעיל, כל סימן פלוס (+) ניתן להחלפה בנקודה־פסיק.

CHR \$ 22 (בקר AT) מלווה בשני

ערכי CHR \$ לציון מספרי השורה ועמודה. הפקודה

**PRINT CHR \$ 22 + CHR \$ 11 +  
CHR \$ 16 + CHR \$ 42**

מציגה כוכב במרכז המסך.

CHR \$ 23 (בקרת TAB) מלווה גם היא בשני ערכים באותה צורה. הערך השני הוא בד"כ 0 והראשון נותן את מיקום TAB. הפקודה

**PRINT CHR \$ 23 + CHR \$ 16 +  
CHR \$ 0 + CHR \$ 42**

מציגה כוכב במחצית הדרך לרוחב המסך.

שים לב לכך שקיימים רק בקרים אלה.

שימוש ב־PRINT CHR \$ עם ערך של מילת מפתח גדול מ־164 מציג רק את מילת המפתח אך לא מפעיל אותה.

פורמט

CHR \$ int-num-const [;] [+]  
CHR \$ int-num-var [;] [+]  
CHR \$ int-num-expr [;] [+]

**CIRCLE**

מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT H

הצהרה/פקודה

CIRCLE מצוירת עיגול על המסך.

**כיצד להשתמש ב־CIRCLE**  
CIRCLE מלווה ב־3 ערכים מספריים המופרדים בפסיקים, לדוגמה

**80 CIRCLE x,y,z**

כל אחד משלשת הערכים מעוגל לשלם הקרוב ביותר במידת הצורך. CIRCLE מצויר עיגול על רשת הנקודות בהפרדה גבוהה בצבע הדיו הקיים.



CLS

## CODE

מיקום במקלדת

EXTEND MODE

1

פונקציה

CODE נותנת את מספר הקוד של תו במערכת התווים של הספקטרום (ראה עמ' 51).

כיצד להשתמש ב-CODE  
CODE מלווה בערך מחרוזת, לדוגמה

```
90 IF CODE < 65 OR CODE < 90 THEN GOTO 80
```

בטוי מחרוזת חייב להיות בתוך סוגריים. CODE מחזירה את מספר הקוד של התו הראשון במחרוזת. אם זוהי מחרוזת ריקה (""), אזי CODE מחזירה 0.

LOAD CODE ו-SAVE CODE  
השימוש ב-CODE עם LOAD ו-SAVE הוא בצורה שונה. ראה הסברים על LOAD CODE ו-SAVE CODE.

פורמט

CODE string-const  
CODE string-var  
CODE (string-expr)

## CONTINUE

מיקום במקלדת

C

פקודה

אם תכנית נעצרת, ניתן להשתמש ב-CONTINUE להתחיל מחדש את התכנית מהנקודה בה היא נעצרה. אם אירעה שגיאה שגרמה להפסקת התכנית, אזי חייבים לתקנה לפני ש-CONTINUE תאפשר לתכנית להמשיך.

כיצד להשתמש ב-CONTINUE  
CONTINUE משמשת כפקודה ישירה כאשר תכנית נעצרה. אין צורך בנתונים נוספים. לאחר CONTINUE התכנית ממשיכה באופן רגיל באותה הצורה בה היא נעצרה. אם הסיבה הייתה שגיאה, אזי ניתן להקיש פקודה לתיקון השגיאה ו-CONTINUE תאפשר לתכנית להמשיך מההצורה. אם התכנית נעצרה בהצורה STOP במתן דווח 9 או אם נעצרה עקב הקשת BREAK וקבלת דווח 1, אזי CONTINUE גורמת לתכנית להמשיך מההצורה הבאה. ניתן להכניס תחילה פקודה מתקנת, במידת הצורך. אם

## CLEAR 65267

במקרה זה CLEAR מבצעת מה שתואר לעיל וגם מביאה את RAMTOP, הכתובת הגבוהה ביותר של מרחב מערכת בייסיק, לערך הנתון. RAMTOP נקבעת ב-65367 בספקטרום ZX+ ונמצאת מתחת לאיזור השמור עבור גרפיקה בהגדרת-משתמש. NEW מנקה את הזכרון עד ל-RAMTOP ולכן, השימוש ב-CLEAR להנמכת מיקום RAMTOP (ב-100 בתים בדוגמה שלעיל) מאפשר יותר זכרון החסין בפני NEW. העלאת RAMTOP נותנת מרחב גדול יותר לבייסיק על חשבון גרפיקה בהגדרת-משתמש. שים לב לכך שמחסנית GOSUB תמוקם אזי ב-RAMTOP, בתנאי ש-RAMTOP נשארת מעל מחסנית המחשבון. כדי למצוא את RAMTOP, הקש

```
PRINT PEEK 23730 + 256 *  
PEEK 23731
```

פורמט

CLEAR (num-expr)

## CLOSE #

פקודת טיפול בקבצים בכונן זעיר. ראה מדרוך שימוש בכונן זעיר ומימשק 1.

## CLS נקה מסך

מיקום במקלדת

V

פקודה/הצורה

CLS מוחקת את כל הטקסט והגרפיקה משטח התצוגה, ומוחירה אותו נקי בצבע הנייר (רקע) הקיים.

כיצד להשתמש ב-CLS  
ניתן להשתמש ב-CLS כפקודה ישירה או בתור הצורה בתכנית. לא נדרשים כל נתונים נוספים, לדוגמה

```
250 IF & = "NO" THEN CLS
```

שטח התצוגה (אך לא הגבולות) מתנקה ועובר לצבע שנבחר על-ידי הצהרת PAPER או פקודה הקודמת או לצבע הנייר בברירת המחדל, הלבן. שים לב שחייבים להשתמש ב-CLS לאחר PAPER ולפני PRINT או כל הצהרת תצוגה אחרת ליצירת רקע צבעוני על פני כל שטח התצוגה.

שני הערכים הראשוניים (x,y) מגדירים את הקואורדינטות האפקט והאנכית של המרכז, והערך השלישי (z) מגדיר את אורך הרדיוס. המימדים חייבים להיות כאלה שלא יחרטו משטח התצוגה.

CIRCLE מושפעת מהצהרות צבע או פקודות ועשויה לכלול הצהרות צבע מוטמעות עם אותן תוצאות כמו DRAW PLOT

דוגמה  
הפקודה

CIRCLE 128,88,87

מציירת מעגל התופס את מרבית שטח התצוגה.

פורמט

CIRCLE (statement , ) int-num-expr ,  
int-num-expr , int-num-expr

## CLEAR

מיקום במקלדת

X

הצהרה/פקודה

CLEAR מוחקת את הערכים הקיימים של כל המשתנים, משחררת את מרחב הזכרון שהערכים תפסו ומרחב עד מחיצת RAMTOP בזכרון, שהיא הכתובת העליונה במרחב מערכת בייסיק.

ניתן גם להשתמש ב-CLEAR כדי לאפס את RAMTOP.

כיצד להשתמש ב-CLEAR  
ניתן להשתמש ב-CLEAR כפקודה ישירה או בתור הצורה בתכנית. לא נדרשים כל ערכים, לדוגמה

50 CLEAR

במקרה זה CLEAR תמחק את כל הערכים המוקצים לכל המשתנים, כולל מערכים. כמו כן היא תבצע CLS ו-RESTORE במטרה לנקות את המסך ולהחזיר את סמן הנתונים לפריט המידע הראשון. בנוסף, מיקום PLOT מוחזר לפינה שמאלית תחתונה של שטח התצוגה ומחסנית GOSUB מתרוקנת.

שים לב שלא נדרשת פקודת CLEAR לפני הגדרת מימדי מערכים מחדש מפני ש-DIM מוחקת מערך הקיים באותו שם. שים לב גם לכך ש-RUN מבצעת CLEAR.

CLEAR ו-RAMTOP  
ניתן ללוות את CLEAR בערך מספרי, לדוגמה



משתמשים ב-CONTINUE להמשך פקודה ישירה, אזי היא תכנס ללולאה אם הפקודה נעצרה בהצורה הראשונה שבה התצוגה נעלמה, אך ניתן להחזיר את הבקרה בלחיצת BREAK. CONTINUE נותנת דווח 0 אם הפקודה נעצרה בהצורה השנייה ודווח N בהצורה השלישית או בהצורות הבאות.

**פורמט**

CONTINUE

**COPY**

מיקום במקלות

Z

**פקודה**

COPY גורמת למדפסת מסוג סינקלייר להפיק עותק מודפס של תצוגת המסך.

כיצד להשתמש ב-COPY משמשת בתור פקודה ישירה כאשר תכנית הסתיימה או נעצרה. היא לא דורשת כל נתונים. לאחר COPY, ובתנאי שהמדפסת מחוברת, יודפס עותק של 22 השורות הראשונות במסך. שים לב לכך שכל צבעי הדיו מודפסים בשחור; צבעי הנייר (רקע) אינם מודפסים. ניתן לעצור את המדפסת בלחיצת BREAK. אם מופיע רישום תכנית על המסך, ניתן להדפיסו ע"י COPY בתנאי שנוצר בפקודת או הצהרת LIST. שים לב לכך שהקשת ENTER לאחר שתכנית הסתיימה או נעצרה תגרום להופעת רישום על המסך, אך לא ניתן להדפיס רישום "אוטומטי" זה באמצעות COPY.

**פורמט**

COPY

**COS קוסינוס**

מיקום במקלות

EXTEND MODE  
W

**פונקציה**

COS נותנת את קוסינוס הזווית.

כיצד להשתמש ב-COS COS מלווה בערך מספרי, לדוגמה

140 LET X = COS Y

בטוי חייב להיות בתוך סוגריים. הערך המלווה את COS הוא הזווית ברדיאנים. COS מחזירה את קוסינוס הזווית. ניתן להפוך מעלות לרדיאנים ע"י הכפלה ב-PI/180. שים לב ש-COS מחזירה ערך שלילי עבור זוויות בתחום 90 עד 270 מעלות וערך חיובי עבור זוויות בתחום 0 עד 90 ו-270 עד 360 מעלות.

**דוגמה הפקודה**

PRINT COS (60 \* PI/180)

מציגה 0.5, קוסינוס זווית בת 60 מעלות.

**גרסה**

COS num-const  
COS num-var  
COS (num-expr)

**DATA**

מיקום במקלות

EXTEND MODE  
D

**הצהרה**

DATA מטפלת רשימת פריטי מידע בתוך תכנית. פריטים אלה יכולים להיות, לדוגמה, ערכי משתנים או מחרוזות שיש להציג כל פריט מוקצה לטובת משתנה ע"י הצהרת READ. ההקצאה מתבצעת בסדר בו מופיעים פריטי המידע בתכנית, אך ניתן להשתמש ב-RESTORE להתחלת הקצאה בפריט הראשון שבהצהרת DATA נתונה.

כיצד להשתמש ב-DATA ניתן להשתמש ב-DATA ליצירת הצהרה בתכנית בלבד. בר"כ היא מלווה ברשימת קבועים מספריים או מחרוזות, המופיעים כולם בפסיקים, לדוגמה

50 DATA 31, "JAN", 28, "FEB"

כל קבוע מוקצה במקרה זה למשתנה ע"י הצהרת READ הקוראת את DATA. הצהרת DATA ניתנת למיקום בכל מקום בתכנית. המספר, סוג (מספרי או מחרוזת) וסדר הקבועים חייב להתאים למספר הפעמים בו מתבצעת הצהרת READ וסוג וסדר המשתנים בהצהרת READ. ניתן לפצל את רשימת המידע לכמה הצהרות DATA רצופות במקרה של פריטים רבים מדי בהצהרה אחת.

**דוגמה התכנית הבאה:**

```
10 FOR n = 1 TO 2
20 READ x, n$
30 PRINT n$, x, "days"
40 NEXT n
50 DATA 31, "JAN", 28, "FEB"
```

**מציגה**

JAN 31 days  
FEB 28 days

**שימוש ב-DATA עם משתנים**

פריטי המידע בהצהרת DATA עשויים לכלול משתני מחרוזת או מספריים, או ביטויים בתנאי שלמשתנים הוקצו מקומם ערכים. בדוגמה שלעיל, ניתן לשנות את הצהרת DATA בצורה הבאה:

50 DATA n\$, d-3, "FEB"

אם הוקצה מקומם ל-n ערך של 31 ו-d המחרוזת "JAN", תושג אותה התצוגה על המסך.

**SAVE DATA, LOAD DATA, VERIFY DATA**

DATA ניתנת לשימוש גם עם LOAD, SAVE, VERIFY לצורך איחסון מערכים על סרט. ראה LOAD DATA, VERIFY SAVE DATA.

**גרסה**

DATA num-expr {, num-expr}  
[, string-expr]  
DATA string-expr {, num-expr}  
[, string-expr]

**DEF FN הגדר פונקציה**

מיקום במקלות

EXTEND MODE  
SYMBOL SHIFT 1

**הצהרה**

DEF FN מאפשרת למשתמש להגדיר פונקציה שאינה קיימת כמילת מפתח. ניתן להעביר מבחר גדול של נתונים לפונקציה בהצהרת FN, אשר קוראת לפונקציה ועשויה להחזיר ערך מספרי או מחרוזת בתור תוצאה.

כיצד להשתמש ב-DEF FN ניתן להשתמש ב-DEF FN בהצהרה בתכנית בלבד. אם יש להגדיר פונקציה מספרית, DEF FN מלווה במילה בודדת כלשהי ואח"כ במשתנה מספרי אחד או יותר, כשכל אחד



(2)  $x$  זהה ל- $x(2)$ . ברם, משתנים מספריים פשוטים בעלי אותה אות כמו שם מערך ( $x$  או  $x$ ) יכולים להתקיים יחד וניתנים לשימוש בנפרד, במידת הצורך.

מספר הערכים בסוגריים שווה למספר המימדים הנוצרים במערך מספרי. הדוגמה השנייה מכינה מערך דרמימדי בן 100 איברים עם 20 איברים במימד הראשון 5 בשני. איברים אלה ממוספרים  $x(1,1)$  עד  $x(20,6)$ . ניתן ליצור מערכים בכל מספר מימדים.

איברי מערך מספרי ניתנים לזיהוי לפי הסדר ע"ש המערך בלוויית עוד בסוגריים, לדוגמה

```
70 PRINT x(a)
100 PRINT z(7, b)
```

#### DIM ומערכי מחרוזת

ניתן להשתמש ב-DIM באותה צורה כמו עם מערכים מספריים פרט לכך שעבור שם המערך משתמשים באות בודדת המלווה בסימן \$.

כמו כן, יש להוסיף עוד נוסף לערכי המימדים בסוגריים במטרה להגדיר אורך כל מחרוזת, לדוגמה

```
30 DIM a$(20, 5)
90 DIM b$(20, 5, 10)
```

ההצהרה הראשונה יוצרת מערך בן 20 איברים, כשכל אחד מכיל מחרוזת בת 5 תווים. המשתנים המותגים נקראים  $a$(1)$  עד  $a$(20)$  כולל, ובתחילה מוקצה להם מחרוזת אפס (ריקה)  $""$ . כל מערך קיים באותו שם נמחק, ולא כמו במערכים מספריים, לא יכול להתקיים משתנה מחרוזת פשוט נוסף באותו שם.

הדוגמה השנייה יוצרת מערך מחרוזת דרמימדי בן 100 איברים עם 20 איברים במימד הראשון 5 במימד השני. כל האיברים באורך של 10 תווים. כאשר ערכי מחרוזת מוקצים לפי הסדר למערך מחרוזת, הם "מרופדים" במרווחים בסוף המחרוזת או מתקצצים לאורך המוגדר, במידת הצורך.

ביטוי המחרוזת הבא לאחר סימן השוויון בדוגמה זו הוא פורס מחרוזת,  $x$  ו- $y$  הם התו הראשון והתו האחרון של חלק מ- $a$(6)$ . חייב להיות מלווה בשם הפונקציה, וערך מחרוזת יחד עם כל נתון שיש להעבירו לפונקציה בתוך סוגריים. במקרה זה, הפקודה

```
PRINT FN a$(
("FUNDAMENTAL", 1, 3))
```

מציגה FUN, ואילו הפקודה

```
PRINT FN a$(
("FUNDAMENTAL", 5, 8))
```

מציגה AMEN.

#### DEF FN

```
DEF FN letter ([letter]) ([ , letter])
= num-expr
DEF FN letter $ ([letter $]) ([letter]
[ , letter] [ , letter $]) = string-expr
FN letter ([num-expr]) ([ num-expr])
FN letter $ ([string-expr]) ([num-expr]
[ , num-expr] [ , string-expr])
```

#### DIM מימד

מיקום במקלות

D

#### הצהרה

DIM משמשת למתן מימדים (הכנה) למערך בעל מספר נתון של משתני מספרים או מחרוזות. מערך הוא רשימת משתנים באותו שם אשר מזוהים ע"י תגים (ערכים אשר מזוהים כל משתנה או איבר במערך).

כיצד להשתמש ב-DIM עם מערכים מספריים

DIM משמשת ליצירת הצהרה בתכנית. היא מלווה באות בודדת הנותנת שם למערך, ואחד או יותר ערכים מספריים המופרדים בפסיקים, ובתוך סוגריים, לדוגמה

```
20 DIM x(10)
80 DIM z(20, 5)
```

במקרה הראשון, נוצר מערך מספרי דרמימדי המכיל 10 איברים עם תגים מ"ו עד 10. שם המערך הוא  $x$  והמשתנים המותגים הם  $x(1)$  עד  $x(10)$  כולל. כל מערך הקיים באותו שם נמחק, ולמשתנים מוקצה ערך 0. שים לב לכך שבמתן מימדים למערך, הספקטרום אינו מבחין בין שמות עם אותיות גדולות וקטנות – משתנה

מופרד בפסיק וגמצאים בתוך סוגריים, לדוגמה  $DEF FN r(x,y)$  כל זה מלווה בסימן שוויון ואחריו בטוי מספר המכיל את המשתנים, לדוגמה

```
1000 DEF FN r(x, y) =
SQN (x^2 + y^2)
```

האות המלווה את DEF FN ( $r$  שלעיל) היא שם זיהוי לפונקציה. גם המשתנים יכולים להיות רק אותיות בודדות. שים לב לכך שבשני המקרים, הספקטרום אינו מבחין בין אותיות גדולות וקטנות. הביטוי שאחרי סימן השוויון משתמש במשתנים ( $x$  ו- $y$  שלעיל) להגדרת הפונקציה.

ניתן למקם הצהרת DEF FN בכל מקום בתכנית. כדי לקרוא לפונקציה המוגדרת, משתמשים בהצהרת FN. ההצהרה תלווה באות שם הפונקציה ורשימת ערכים מספריים בתוך סוגריים ומופרידים בפסיקים, לדוגמה

```
50 PRINT FN r(3, 4)
```

הערכים שבתוך הסוגריים מועברים לפונקציה באותו סדר כמו המשתנים בהצהרת DEF FN. לכן, בדוגמה זאת, לא מוקצה ערך של 3 ול-4 ערך של 4. FN מחשבת את הביטוי ומחזירה את הערך.

ניתן גם ללוות את DEF FN באות וצמד סוגריים בלבד, לדוגמה

```
1000 DEF FN r() = INT(x + 0.5)
```

הערך המוקצה כעת למשתנה ( $x$  לעיל) מועבר לפונקציה כאשר היא נקראת ע"י FN. במקרה זה,  $r()$  FN מחזירה את הערך המוקצה כעת ל- $x$  כשהוא מעוגל לשלם הקרוב ביותר.

#### DEF FN ומחרוזות

ניתן להשתמש ב-DEF FN באותה דרך גם כדי להגדיר ולקרוא לפונקציית מחרוזת. במקרה זה, שם הפונקציה הוא אות בודדת בלוויית \$ ומשתנה אחד או יותר בפונקציה הוא אות בלוויית \$.

ביטוי מחרוזת מתאימים יוצרים את ההגדרה, לדוגמה

```
1000 DEF FN a$(b$, x, y) =
b$ (x TO y)
```



## דוגמה הפקודה

PRINT EXP 1

מציגה 2.7182818, הערך של e.

## פורמט

EXP num-const  
EXP num-var  
EXP (num-expr)

## FLASH

### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT V

### הצהרה/פקודה

FLASH גורמת להבהוב במיקומי התווים, בכך שצבעי הדיו והנייר מתחלפים בקצב קבוע.

כיצד להשתמש ב־FLASH  
ניתן להשתמש ב־FLASH כפקודה ישירה אך בד"כ היא יוצרת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

50 FLASH 1

הערך הבא אחרי FLASH מעוגל לשלם הקרוב ביותר במידת הצורך ועשוי להיות 0, 1 או 8. ערך של 1 גורם לכל התווים שיוצגו בהמשך ע"י PRINT או INPUT להבהב. ערך של 8 גורם למיקומי התווים המהבהבים להמשיך ולהבהב, ולמיקומי תווים רגילים להישאר במצבם גם לאחר הכנסת תווים חדשים. FLASH בלוחית 0 מבטלת את 1 FLASH וגם את 8 FLASH כך שכל התווים שיוצגו בהמשך הם רגילים. ניתן גם להטמיע (להכניס) FLASH בתוך הצהרות תצוגה הנוצרות ע"י PRINT, INPUT, PLOT, DRAW ו־CIRCLE. FLASH מלווה את מילת המפתח אך מקדים את נתוני המידע או התצוגה; הוא מלווה באותם ערכים ונקודה פסיק, לדוגמה

120 PRINT FLASH 1 ; INK 2 ;  
PAPER 6 ; "WARNING"

ל־FLASH יש, איפוא, השפעה מקומית הישימה רק לתווים המוצגים, נקודות מותוות או קוים שצוירו בהצהרת תצוגה. שים לב לכך ש־1 FLASH גורמת לכל 8x8 מיקומי הפיקסלים להבהב אם אחד מהפיקסלים מותווה בצבע דיו.

DRAW מושפעת מהצהרות או פקודות צבע ועשויה לכלול הצהרות מוטמעות עם אותם האפקטים כמו עם PLOT ו־CIRCLE.

ציור קוים עקומים  
ניתן להוסיף ל־DRAW ערך שלישי ליצירת עקומה המהווה חלק מעיגול, לדוגמה

40 DRAW x, y, z

הערך השלישי (z שלעיל) מגדיר את הזווית (ברדיאנים) שאותה עובר הקו תוך כדי ציורו. הקו מסתובב שמאלה אם הערך חיובי, וימינה אם הערך שלילי. ערכים של 0 או 0.1 יוצרים עיגול.

## דוגמה

התכנית הבאה מציירת משולש:

10 PLOT 127, 150  
20 DRAW 70, -100  
30 DRAW -140, 0  
40 DRAW 70, 100

הוספת 1 או 1- ליצירת DRAW גורמת לצדדים להתעקם פנימה או החוצה, בהתאמה.

## פורמט

DRAW [statement ; ] int-num-expr  
int-num-expr { , int-num-expr }

## ERASE

פקודת טיפול בקבצים בטון זעיר.  
ראה מדריך כונן זעיר ומימשק 1.

## EXP אקספוננט

### מיקום במקלדת

EXTEND MODE  
x

### פונקציה

EXP היא פונקציה מתימטית המעלה את האקספוננט e בחזקה הנתונה.

כיצד להשתמש ב־EXP  
EXP מלווה בערך מספרי, לדוגמה

60 LET y = EXP x

ביטוי חייב להיות בתוך סוגריים. EXP מחזירה את האקספוננט e כשהוא מועלה בחזקת הארגומנט (x לעיל).

איברי מערך מחרוזת מוזהים ע"י שם המערך המלווה, בסוגריים, באחד או יותר ערכים מספריים הנותנים את מספר(י) התג. לדוגמה, איבר as(2) עשוי להיות "SMITH" ואיבר bs(12,4) עשוי להיות "DERBYSHIRE". ברם, ניתן להוסיף ערך נוסף להגדרת תו מסויים במחרוזת. בדוגמאות אלה, as(2,2) יהיה "M" (התו השני ב־"SMITH") ו־bs(12,4,5) יהיה "Y".

ערכי מחרוזת במימד אפס  
ניתן ליצור מערך מחרוזת במימד אפס ע"י שימוש בערך אחד בלבד בסוגריים, לדוגמה

10 DIM c\$(15)

למערך זה יש איבר אחד בלבד, שהוא c\$, וארכו קבוע בערך המוגדר (15 תווים).

## פורמט

DIM letter (num-expr { , num-expr })  
DIM letter\$ (num-expr { , num-expr })

## DRAW

### מיקום במקלדת

w

### הצהרה/פקודה

DRAW משמשת לציור קוים ישרים ועקומות על המסך.

כיצד להשתמש ב־DRAW  
DRAW משמשת בד"כ ליצירת הצהרה בתכנית. אם נדרש קו ישר, היא מלווה בשני ערכים מספריים המופרדים בפסיק, לדוגמה

40 DRAW x, y

קו ישר יצויר על רשת נקודות הגרפיקה בהפרדה גבוהה מתנקודה המוגדרת ע"י הצהרת PLOT הקודמת או הנקודה אליה הגיעה הצהרת DRAW הקודמת, מה שבא אחרון. שני הערכים המלווים את DRAW מעוגלים לשלם הקרוב ביותר, אם נדרש.

הערך הראשון (x לעיל) מגדיר את המרחק האפקי מנקודה זאת, והערך השני (y) את המרחק האנכי. ערכים אלה הם שליליים אם הקו פונה שמאלה או מטה בהתאמה, ונקודת היעד חייבת להימצא בגבולות שטח התצוגה.

אם לא קיימת הצהרת PLOT או DRAW קודמת, DRAW מתחילה מנקודה 0,0 (פינה שמאלית תחתונה במסך).



FLASH int-num-expr[, ]

**FN פונקציה****מיקום במקלות**EXTEND MODE  
SYMBOL SHIFT 2**פונקציה**

FN קוראת לפונקציה בהגדרת משתמש. היא בשימוש תמיד בקשר עם DEF FN, אשר מגדירה את הפונקציה הנקראת.

**מיצד להשתמש ב-FN**

אם יש לקרוא לפונקציה מספרית, FN מלווה באות ואחריה זוג סוגריים. אם יש להעביר נתונים מסוימים לפונקציה, הם יוכנסו, מופרדים בפסיקים, בתוך הסוגריים, לדוגמה

170 LET x = FN r(3, 4)

הנתונים (3, 4) שלעיל מועברים לפונקציה הנקראת FN r. מחזירה את התוצאה. אם אין נתונים להעברה, עדיין יש לכלול את צמד הסוגריים, לדוגמה

70 PRINT FN r()

במקרה זה, הפונקציה משתמשת בערכים המוקצים באותו זמן למשתניה. FN קוראת לפונקציית מחזרת באותה צורה, פרט לכך שיש להוסיף סימן \$ אל האות של שם הפונקציה. ראה פרטים נוספים ב-DEF FN

**פורמט**

FN letter ([num-expr] [, num-expr])  
FN letter\$ ([string-expr] [num-expr]  
[, num-expr] [, string-expr])

**FOR****מיקום במקלות**

F

**הצהרה/פקודה**

FOR פועלת תמיד יחד עם מילות המפתח TO ו-NEXT ליצירת לולאת FOR NEXT. מבנה זה מאפשר חזרה על חלק מסויים בתכנית מספר נתון של פעמים.

**מיצד להשתמש ב-FOR**

FOR משמשת תמיד ליצירת הצהרה יחד עם TO. FOR מלווה באות, סימן שוויון, ואחריהם שני ערכים מספריים המופרדים על ידי TO, לדוגמה:

60 FOR a = 1 TO 9

האות (a שלעיל) יוצרת משתנה בקרה. אח"כ באות ההצהרות שיש לחזור על ביצוע, כאשר אחת או יותר מהן מנצלת בד"כ את משתנה הבקרה. הלולאה מסתיימת בהצהרת NEXT, שבה NEXT מלווה במשתנה הבקרה. לדוגמה

90 NEXT a

בעת הביצוע, FOR מוחק את כל המשתנים בעלי אותו שם כמו משתנה הבקרה ומקצה לו ערך ראשוני שיבוא לפני TO (1 לעיל). ההצהרות מתבצעות כאשר משתנה הבקרה מכיל ערך זה. כאשר מגיעים ל-NEXT, ערך משתנה הבקרה עולה ב-1. אם ערך זה קטן או שווה לערך שאחרי TO (ערך הגבול של 9 לעיל), התכנית חוזרת להצהרת FOR ולולאת FOR NEXT חוזרת. אם משתנה הבקרה מקבל ערך גדול מערך הגבול, הלולאה מסתיימת והתכנית ממשיכה בהצהרה הבאה אחרי NEXT בדוגמה שלעיל, הלולאה חוזרת 9 פעמים כשמשתנה הבקרה משתנה מ-1 עד 9. ביציאה מהלולאה, a הוא בערך של 10. שים לב לכך שהספקטרום אינו מבחין בין אותיות גדולות וקטנות בעת קריאת שם למשתנה הבקרה.

שימוש ב-STEP בלולאת FOR NEXT  
STEP היא מילת מפתח אשר ניתן לשלבה בהצהרת FOR אם יש להגדיל את משתנה הבקרה בערך שונה מ-1 או להפחיתו.  
STEP מלווה את ערך הגבול ומלווה בערך מספרי, לדוגמה

60 FOR a = 1 TO 9 STEP 2

משתנה הבקרה מוגדל בערך המדרגה (2 לעיל) עד שהוא גדול מערך הגבול. משתנה הבקרה מקבל ערכים עוקבים של 1, 3, 5, 7, 9 ועוב את הלולאה בערך של 11.

ערך מדרגה שלילי גורם להפחתת ערך משתנה הבקרה. במקרה זה, הערך הראשוני חייב להיות גדול מזה של ערך הגבול והלולאה מסתיימת כאשר ערך משתנה הבקרה נמוך מזה של ערך הגבול, לדוגמה

60 FOR a = 9 TO 1 STEP -1

ערך a יורד מ-9 עד 1, a עוב את הלולאה כאשר ערכו 0.

**לולאות מקוננות**

ניתן למקם אחת או יותר לולאות FOR NEXT אחת בתוך השנייה, תהליך הנקרא "קינן" לולאות. סדר משתני הבקרה בהצהרות NEXT חייב להיות בסדר הפוך למשתני הבקרה בהצהרות FOR. לולאות FOR NEXT ניתנות לקינון בכל עומק, כלומר, ניתן למקם לולאות אחת בתוך השנייה ככל שגדרש.

**פורמט**

FOR letter = num-expr TO num-expr  
[STEP num-expr] NEXT letter

**FORMAT**

פקודת טיפול בקבצים בכונן זעיר. ראה מדרוך כונן זעיר ומימשק 1.

**GOSUB****מיקום במקלות**

H

**הצהרה/פקודה**

GOSUB גורמת לתכנית להסתעף אל שיגרה, שהיא קטע נפרד בתכנית. הפעולה שימושית אם השיגרה נדרשת מספר פעמים בתכנית.

מיצד להשתמש ב-GOSUB  
ניתן להשתמש ב-GOSUB כהצהרה או פקודה ישירה והיא מלווה בערך מספרי, לדוגמה

GOSUB 1000

בזמן הביצוע, הערך המלווה את GOSUB (1000 שלעיל) מעוגל לשלם הקרוב ביותר, והתכנית מסתעפת למספר השורה בעלת אותו ערך. השימוש במשתנה או בטוי מאפשר לתכנית להסתעף אל שיגרה במספר שורה מחושב. שים לב לכך שאם מספר השורה לא קיים, התכנית עדיין מסתעפת וממשיכה עם ההצהרה הראשונה שמגיעים אליה. כל שגרה מסתיימת ב-RETURN, והתכנית מסתעפת חזרה אל ההצהרה הבאה אחרי הצהרת GOSUB. ניתן לקנן שגרות כך שמשיגרה אחת מגיעים לשנייה, ובמקרה זה RETURN שולח את התכנית חזרה להצהרה הבאה אחרי הצהרת GOSUB האחרונה שבוצעה.



## מחסנית GOSUB

כאשר מתבצעת GOSUB, מספר השורה מועבר למחסנית GOSUB בזכרון. אם מתבצעות שתי הצהרות GOSUB או יותר לפני RETURN, מספרי השורות שלהן נעדמים במחסנית כך שהמספר האחרון נמצא בראש המחסנית. RETURN לוקחת תמיד את מספר השורה מראש המחסנית ופונה לשורה זו להמשך התכנית.

שים לב לכך שתיתכן שגיאה 4 (Out of memory) במידה ואין מספיק הצהרות RETURN.

## פורמט

GOSUB int-num-expr.

## GOTO

## מיקום במקלדת

## הצהרה/פקודה

GOTO גורמת לתכנית להסתעף אל שורה מסוימת.

**כיצד להשתמש ב-GOTO**  
ניתן להשתמש ב-GOTO כפקודה ישירה להרצת תכנית ממספר שורה נתון ללא ניקוי המסך תחילה. ניתן להשתמש בה גם ליצירת הצהרה בתכנית.

GOTO מלווה בערך מספרי, לדוגמה

60 GOTO 350

בעת הבצוע, הערך המלווה את GOTO מעוגל לשלם הקרוב ביותר והתכנית מסתעפת למספר השורה באותו ערך. השימוש במשתנה או בטוי מאפשר לתכנית להסתעף אל מספר שורה מחושב. שים לב לכך שאם השורה לא קיימת, התכנית עדיין תסתעף ותמשיך בהצהרה הראשונה שתמצא.

## פורמט

GOTO int-num-expr

## הצהרה/פקודה

IF באה לדירב בצירוף מילת המפתח THEN כדי לפנות להחלטה המשפיעה על הפעולות הבאות. כדי לבצע זאת,

המחשב בודק משהו כדי למצוא האם הוא אמת או לא. אם הוא אמת, יופעל נתיב פעולה מסוים. אם הוא לא אמת, יופעל נתיב אחר.

**כיצד להשתמש ב-IF THEN**  
IF יוצר בדרך כלל הצהרה עם THEN IF מלווה בתחילה בערך מספרי או בתנאי, ואח"כ ב-THEN עם הצהרת ביסיק אחת או יותר, לדוגמה

80 IF x THEN GOTO 250  
240 IF a\$ = "NO" THEN PRINT  
"THE END" : STOP

קבוע, משתנה או בטוי (כמו x לעיל) נחשב אמת אם ערכו שונה מאפס. במקרה זה, מתבצעות ההצהרה הבאה אחרי THEN וכל הצהרה נוספת באותה שורה. התכנית ממשיכה לשורה הבאה. אם הערך הוא 0, אזי הקבוע, משתנה או בטוי נחשב שקרי. ההצהרות הבאות לא מתבצעות והתכנית מדלגת לשורה הבאה. בדוגמה זו, התכנית לא תפנה לשורה 250 אם x שווה 0.

אם התנאי ("a\$ = "NO") הבא אחרי IF הוא אמת, אזי מתבצעות ההצהרות הבאות אחרי THEN. אם התנאי שקרי, התכנית תפנה לשורה הבאה. בדוגמה זו, אם a\$ הוא בעל ערך "NO" אזי מוצג "THE END" והתכנית נעצרת. אם ל-a\$ יש ערך אחר כלשהו, התכנית תמשיך מהשורה הבאה. הספקטרום נותן לתנאי אמת ערך של 1 ולתנאי שקרי ערך 0. הוא מכיר כל ערך שונה מאפס כאמת וכל ערך 0 כשקרי. ניתן להקצות למשתנה את ערך התנאי בעזרת הצהרה, כמו

70 LET x = a\$ - "NO"

שים לב לכך, שלא כמו בנייבי ביסיק אחרים, לא ניתן להשמיס את THEN לפני GOTO.

## פורמט

IF num-expr THEN statement  
[ statement]  
IF cond THEN statement  
[ statement]

## IN

## מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT I

## פונקציה

IN בודקת את מצב המקלדת והתקני קלט ופלט אחרים. היא קוראת בית מכתובת שער נתונה אשר מציינת את מצב ההתקן המחובר לשער.

**כיצד להשתמש ב-IN**  
IN מלווה בערך מספרי, לדוגמה

150 LET x = IN y

הערך המלווה את IN נמצא בתחום 0 עד 65535 ומגדיר את כתובת השער שיש לקרוא. IN מחזירה את הבית הנקרא משער זה.

## כתובות מקלדת

למקלדת יש 8 כתובות, שכל אחת מהן עשויה להכיל אחד מתוך 5 בתים שונים, בהתאם למקש הנלחץ. הכתובות הן 65278, 65022, 64510, 63486, 61438, 57342, 49150, 32766. ערכי בית בכתובות אלה עשויים להיות 175, 183, 187, 189 או 190.

## פורמט

IN num-const  
IN num-var  
IN (num-expr)

## INK

## מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT X

## הצהרה/פקודה

INK מגדירה את הצבע העליון בו מוצגים התאים, הנקודות המוחות וקווים ועקומות מצוירים.

**כיצד להשתמש ב-INK**  
ניתן להשתמש ב-INK כפקודה ישירה אך בד"כ היא משמשת ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

70 INK x

הערך המלווה את INK מעוגל לשלם הקרוב ביותר ועשוי להמצא בתחום מ-0 עד 9. הצבעים העליונים הם כלהלן:  
■ שחור  
1 כחול



```
80 INPUT INK 2 : "What is your
name?" : n$ : ("How old are you , "
+ n$ + "?" ) : age
```

שים לב להבדלים הבאים לעומת  
INPUT, PRINT ממתינה כאשר היא  
מגיעה אל משתנה, ולכן כל המשתנים  
וביטויים (כמו אלה הכוללים n\$ לעיל)  
העתידיים להיכלל בדברונים חייבים  
להיות בתוך סוגריים. התצוגה  
מתחילה בתחילת השורה התחתונה  
ומגולגלת מעלה אם יש יותר משורה  
אחת. ניתן להשתמש ב-AT בתוך  
הצהרת INPUT באותה צורה כמו עם  
PRINT. AT 0,0 מציגה בתחילת השורה  
מעל השורה התחתונה והתצוגה  
מגולגלת כלפי מעלה אם מוצגות יותר  
משתני שורות.

**כיצד לעצור INPUT**  
אם INPUT מלווה בערך מספרי ומוקש  
STOP, התכנית תיעצר. עם משתנה  
מחרת, ניתן למחוק את סימן  
המרכאות הראשון ואז להכניס STOP  
כדי לעצור את התכנית.

**שימוש ב-INPUT עם LINE**  
ניתן להשתמש ב-INPUT LINE עם  
משתני מחרת בלבד. בד"כ, INPUT  
עם משתנה מחרת גורמת להצגת  
צמד מרכאות. כאשר המחרת  
מוקשת פנימה, היא מוצגת בין  
המרכאות. כדי למחוק את המרכאות,  
העזר ב-INPUT LINE מלווה במשתנה  
המחרת. אם נדרש דרבון, הוא  
ממוקם בין INPUT ל-LINE, לדוגמה

```
70 INPUT "What is your
name?" : LINE n$
```

#### פורמט

```
INPUT [prompt] [ , ] [ ' ] num-var
INPUT [prompt] [ , ] [ ' ] string-var
INPUT [prompt] [ , ] [ ' ] LINE
string-var
[prompt] = [string-const]
[(string-expr)] [AT int-num-expr,
int-num-expr] [statement] [ ; ] [ , ] [ ' ]
```

#### INT שלם

##### מיקום במקלדת

EXTEND MODE

■

#### פונקציה

INT משנה מספרים ממשיים (שאינם  
שלמים) למספרים שלמים.

בעת הבצוע, INKEY\$ מחזירה את התו  
הניתן ע"י המקש הנלחץ באותו רגע.  
אם לא נלחץ כל מקש, אזי INKEY\$  
מחזירה מחרתת אפס (ריקה) ("").  
שים לב לכך ש-INKEY\$ מבחינה בין  
אותיות גדולות וקטנות ותוים אחרים  
עיליים (SHIFT) או לא עיליים.  
(העזר ב-IN לזיהוי מקש כלשהו  
ללא תוים מזהים).  
שלא כמו INPUT, INKEY\$ אינה  
מחכה אלא עוברת מייד להצהרה  
הבאה. לכן היא ממוקמת לרוב בתוך  
לולאה החזרת ומתבצעת עד שהמקש  
הנדרש נלחץ.

#### דוגמה

שורה זו משהה את הפעולה עד אשר  
מקש Y נלחץ (ללא CAPS SHIFT או  
CAPS LOCK).

```
60 IF INKEY$ <> "Y" THEN
GOTO 60
```

#### פורמט

INKEY\$

## INPUT

#### מיקום במקלדת

1

#### הצהרה/פקודה

INPUT מאפשרת הכנסת נתונים תוך  
כדי ריצת תכנית.

**מוצד להשתמש ב-INPUT**  
INPUT יוצרת לרוב הצהרה בתכנית  
ומשמשת בצורה דומה ל-PRINT.  
בצורתה הפשוטה ביותר, היא מלווה  
במשתנה מספרי או מחרת, לדוגמה

```
60 INPUT x
90 INPUT a$
```

המחשב ממתיק עד שיוכנס מספר או  
מחרת. הערך מוצג בתחילת השורה  
התחתונה תוך הקשתו פנימה.  
בלחיצת ENTER, הערך מוקצה  
למשתנה הנקרא והתכנית ממשיכה.  
הצהרת INPUT עשויה לכלול יותר  
ממשתנה אחד והיא תציג תוים  
ליצירת סמן דרבון (prompt). זה  
מתבצע בדיוק באותה צורה כמו עם  
PRINT, תוך שימוש בסימני מרכאות  
הסוגריים את תווי סמן הדרבון וסימני  
נקודה פסיק או פסיקים כנודש  
להפרדה בין פריטים.

הצהרות תצוגה כמו INK FLASH ו-  
PAPER ניתנות להטמעה, לדוגמה

2 אדום  
3 סגול  
4 ירוק  
5 תכלת  
6 צהוב  
7 לבן  
8 שקוף

9 צבע ניגודי שחור או לבן.  
8 INK מגדירה שהצבע הקיים נשאר  
ללא שינוי בכל נקודה על המסך  
עבורה הוגדר 8 INK. 9 INK גורמת  
לצבע הדיו להיות שחור או לבן כך  
שיבלוט על רקע צבע הנייר (הרקע).

#### צבעי דיו מקיפים ומקומיים

כאשר INK יוצרת הצהרה לחד,  
כמתואר לעיל, הצבע הוא מקיף וכל  
התצוגות הבאות יופיעו באותו צבע  
עליו. INK עשויה גם להיות מוטמעת  
(מוכנסת) בהצהרות תצוגה הנוצרות  
ע"י DRAW, PLOT, INPUT, PRINT ו-  
INK CIRCLE באה אחרי מילת המפתח  
אך מקדימה את המידע או נתוני  
התצוגה; היא מלווה באותם ערכים  
ונקודה פסיק, לדוגמה

```
60 CIRCLE INK 4 : 128 , 88 , 87
```

ההשפעה של INK היא, איפוא,  
מקומית ושימה רק לתוים המוצגים,  
נקודות מותוות או קווים המצויירים  
ע"י הצהרת התצוגה, ובדוגמה זו ציור  
עיגול ירוק. לאחר מכן צבע הדיו חוזר  
לצבע המקיף או לברירת המחדל –  
שחור.

#### פורמט

INK int-num-expr [ ; ]

## INKEY\$ מחרתת מקש קלט

#### מיקום במקלדת

EXTEND MODE  
N

#### פונקציה

INKEY\$ משמשת לזיהוי לחיצת  
מקשים במקלדת.

**כיצד להשתמש ב-INKEY\$**  
INKEY\$ אינה דורשת כל ארגומנט  
ומשמשת לרוב להקצאת תו עבור  
משתנה מחרת או לבדיקת תו  
מסוים, לדוגמה

```
70 LET a$ = INKEY$
130 IF INKEY$ = "N" THEN STOP
```



כיצד להשתמש ב-INT  
INT מלווה בערך מספרי, לדוגמה

70 LET x = INT y

ביטוי חייב להיות בתוך סוגריים. INT מחזירה את הערך כשהוא מעוגל למספר שלם.

דוגמה  
הפקודה

PRINT INT 45.67, INT -7.66

תוציגה  
-8

45

פורמט

INT num-const  
INT num-var  
INT (num-expr)

## INVERSE

מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT M

הצהרה/פקודה

INVERSE גורמת לצבעים להופיע הפוכים במיקומי תווים כך שצבע הדיו הופך לצבע הנייר ולהיפך.

כיצד להשתמש ב-INVERSE  
INVERSE משמשת לרוב ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

70 INVERSE 1

הערך המלווה את INVERSE מעוגל לשלם הקרוב ביותר והוא עשוי להיות 0 או 1. INVERSE 1 גורם לכל התצוגות הבאות הנוצרות על ידי PRINT ו-INPUT להופיע בצבעים הפוכים. INVERSE מחזירה את צבעי הדיו והנייר למצב הרגיל.

שים לב לכך שניתן להטמיע (להכניס) את INVERSE בתוך הצהרות תצוגה כפי שנעשה עם INK, ברם, כאשר פועלים עם PLOT, CIRCLE או DRAW, INVERSE 1 גורמת לשורה או נקודה להיות מותווים בצבע הנייר כך שהם נעלמים.

פורמט

INVERSE int-num-expr

LEN אוך מחרחת

מיקום במקלדת

EXTEND MODE  
K

פונקציה

LEN נותנת אוך של מחרחת.

כיצד להשתמש ב-LEN  
LEN מלווה בערך מחרחת, לדוגמה

50 LET x = LEN a\$

בטוי חייב להיות בתוך סוגריים. LEN מחזירה את מספר התווים במחרחת.

דוגמה  
השורה הבאה

120 INPUT a\$: IF LEN a\$ > 9  
THEN GOTO 120

מעבירה רק מחרחות המכילות עד תשעה תווים.

פורמט

LEN string-const  
LEN string-var  
LEN (string-expr)

## LET

מיקום במקלדת

L

הצהרה/פקודה

LET משמשת להקצאת ערך למשתנה. בביסוק של סינקליר לא ניתן להשמיט את LET בהצהרת הקצאה.

כיצד להשתמש ב-LET  
LET יוצרת לרוב הצהרה בתכנית אך עשויה לשמש גם כפקודה ישירה. היא מלווה במשתנה מספרי או מחרחת, סימן שוויון, וערך. הערך יכול להיות מספרי או מחרחת, בהתאם למשתנה שלפני LET, לדוגמה

60 LET x = x + 1  
80 LET a\$ = "Correct"

הערך מוקצה למשתנה. שים לב שמשתנים פשוטים אינם מוגדרים עד שהם מקבלים ערכים על ידי READ או INPUT. משתני מערך מצד שני מאותחלים ל-0 או מחרחת אפס (ראה DIM).

מודעות

LET num-var = num-expr  
LET string-var = string-expr

## LINE

מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT 3

ראה SAVE

## LIST

מיקום במקלדת

K

הצהרה/פקודה

LIST יוצרת רישום של התכנית הנמצאת בזכרון.

כיצד להשתמש ב-LIST

LIST משמשת לרוב כפקודה ישירה אך עשויה גם ליצור הצהרה בתכנית. כדי לרשום תכנית שלמה, משתמשים בה בנפרד. לאחר מתן הפקודה הישירה

LIST

מופיע העמוד הראשון ברישום והעמודים הבאים יגולגלו במעלה המסך בלחיצת מקש כלשהו פרט ל-A, מקש הרווחים, STOP או BREAK ניתן גם ללוות את LIST במספר שורה, בצורת ערך מספרי, לדוגמה

LIST 100

הערך המלווה את LIST מעוגל לשלם הקרוב ביותר במידה ונדרש, והרישום מתחיל בשורה זאת. אם לא קיימת שורה עם מספר זה, הרישום מתחיל בשורה הבאה.

פורמט

LIST [int-num-expr]

## LLIST רישום במדפסת

מיקום במקלדת

EXTEND MODE  
V

פקודה/הצהרה

LLIST גורמת למדפסת מסוג סינקליר להפיק תדפיס רישום של התכנית הנמצאת בזכרון.

כיצד להשתמש ב-LLIST

LLIST פועלת באותה צורה כמו LIST (ראה LIST לפרטים נוספים). שים לב שתצוגת המסך אינה משתנה עם הדפסת הרישום.

פורמט

LLIST [int-num-expr]

LN לוגריתם טבעי

מיקום במקלדת

EXTEND MODE  
Z



## פורמט

LOAD string-expr CODE  
[int-num-expr][,int-num-expr]

## LOAD DATA

## מיקום במקלות

J  
EXTEND MODE  
D

## הצהרה/פקודה

LOAD DATA משמשת לטעינת מערכים מסרט. המערכים מוקלטים בעזרת SAVE DATA

כיצד להשתמש ב־LOAD DATA ניתן להשתמש ב־LOAD DATA ליצירת הצהרה בתכנית או כפקודה ישירה. LOAD מלווה בתחילה בשם קובץ, שהוא משתנה מחרוזת, ואחריו DATA בתוספת אות או אות וסימן \$, ולבסוף בצמד סוגריים ריקים, לדוגמה

270 LOAD "numbers" DATA n( )  
300 LOAD "names" DATA n\$ ( )

שם הקובץ המלווה את LOAD הוא השם הניתן למערך על הסרט וכפוף לאותן הגבלות כמו שמות תכניות המשמשים ב־LOAD. האות או אות וסימן \$ בעקבות DATA היא השם הניתן למערך בתכנית כאשר הוא נטען ומופעל.

בעת הבצוע, הספקטרום מחפש את שם המערך על הסרט. כאשר השם נמצא, מופיעה התודעה על מערך מספרי או מערך תוים, בלויית השם, והמערך נטען.

כל מערך הנמצא מקודם בזכרון ונושא אותו שם אות (n או \$n לעיל) נמחק, ונוצר מערך חדש באותו שם. אות כשכל ערכיו מאוחסנים בסרט. שים לב שבמערכי תוים, כל משתנה מחרוזת הנמצא אותו זמן בזכרון ונושא אותו שם אות נמחק אף הוא.

## פורמט

LOAD string-expr DATA letter [\$] ( )

## LOAD SCREEN\$

## מיקום במקלות

J  
EXTEND MODE  
SYMBOL SHIFT K

## LOAD CODE

## מיקום במקלות

J  
EXTEND MODE  
I

## פקודה/הצהרה

LOAD CODE משמשת לטעינת קטע בזכרון בנתונים שאוחסנו על סרט. הנתונים כוללים מערך של בתים, ואלה נשלחים אל מערך כתובות בזכרון. ניתן להשתמש ב־LOAD CODE לטעינת תצוגה, או לטעינת נתונים עבור גרפיקה בהגדרת משתמש, לדוגמה.

כיצד להשתמש ב־LOAD CODE ניתן להשתמש ב־LOAD CODE כפקודה ישירה או ליצירת הצהרה בתכנית. LOAD מלווה בשם קובץ, שהוא ערך מחרוזת, ואחריו CODE, לדוגמה

LOAD "data" CODE

שם הקובץ אחרי LOAD הוא שם הנתונים הנטענים והוא כפוף לאותן הגבלות כמו שם תכנית (ראה LOAD). LOAD CODE מחפשת את שם הנתונים וכאשר הוא נמצא, מוצגים הבתים בלויית שם קובץ הנתונים. הספקטרום טוען את הבתים בזכרון באותן כתובות שמהן נשלחו לאחסון. בתהליך הטעינה עולים על כל הנתונים שהיו רשומים בזכרון מקודם. ניתן גם ללוות את CODE באחד או שני ערכים מספריים, המופרדים בפסיק, לדוגמה

LOAD "picture" CODE  
16384, 6912

הערכים הבאים אחרי CODE מעוגלים לשלם הקרוב ביותר ואז מגדירים את כתובת ההתחלה (16384 שלעיל) בה יש לטעון את הנתונים הנקראים, ואת מספר הבתים (6912) אותם יש לשלוח לכתובות המתחילים באותה כתובת. אם המספר שגוי, ניתן דווח שגיאת טעינת סרט. אם אחרי CODE בא רק ערך אחד, הוא מגדיר את הכתובת הראשונה אשר ממנה והלאה ימוקמו כל הבתים.

הדוגמה שלעיל ניתנת לביצוע גם באמצעות מילות המפתח LOAD SCREEN\$ לפרטים נוספים על אחסון בתים, ראה SAVE CODE

## פונקציה

LN נותנת את הלוגריתם הטבעי (לוגריתם על בסיס e) של ערך. היא פועלת כהפכית של EXP.

כיצד להשתמש ב־LN LN מלווה בערך מספרי, לדוגמה

60 LET x = LN y

בטוי חייב להיות בתוך סוגריים. הערך המלווה את LN חייב להיות גדול מ־0. LN מחזירה את הלוגריתם הטבעי של ערך זה.

## פורמט

LN num-const  
LN num-var  
LN (num-expr)

## LOAD

## מיקום במקלות

J

## פקודה/הצהרה

LOAD טוענת תכנית שלמה אל הזכרון מתוך סרט.

כיצד להשתמש ב־LOAD LOAD משמשת לרוב כפקודה ישירה, אך היא עשויה ליצור הצהרה בתכנית כדי לטעון תכנית חדשה. LOAD מלווה בשם קובץ, שהוא משתנה מחרוזת באורך עד 10 תווים, לדוגמה

LOAD "file name"

בעת הבצוע, התכנית הנמצאת כעת בזכרון, וכל ערכי משתניה, נמחקים. הספקטרום מחפש אחר התכנית ששמה נקרא וטוען אותה לאחר שנמצאה. שים לב שהמחשב מבחין בין אותיות גדולות וקטנות בשמות תכניות.

אם בעקבות LOAD באה מחרוזת אפס, כמו בדוגמה זאת

LOAD " "

אזי הספקטרום טוען את התכנית השלמה הראשונה שהוא מוצא. שים לב שהשימוש ב־LOAD שונה כאשר מחובר כונן זעיר. ראה מדרך כונן זעיר ומימשק ZX1 לפרטים נוספים.

## פורמט

LOAD string-expr



## הצהרה/פקודה

LOAD SCREENS מאפשרת טעינת תצוגת המסך ישירות מהסרט. היא שולחת נתונים מהסרט לקטע בזכרון אשר מבקר את תצוגת המסך במטרה להפיק את התמונה.

**כיצד להשתמש ב־LOAD SCREENS**  
ניתן להשתמש ב־LOAD SCREENS ליצירת הצהרה בתכנית או כפקודה ישירה. LOAD מלווה בשם קובץ, שהוא ערך מחרוזת, ואחריו SCREENS לדוגמה.

## LOAD "picture" SCREENS

שם הקובץ הבא אחרי LOAD הוא השם הניתן לנתוני המסך שעל הסרט, כפוף לאותן הגבלות כמו שמות תכניות הנמצאים בשימוש עם LOAD הספקטרום מחפש את הנתונים ששםם נקרא וכאשר מוצא אותם, הם נטענים ראשית לקובץ התצוגה ואחר כך לקטע המאפיינים בזכרון. התמונה נבנית לאט בצבעי הדיו והנייר הקיימים ואז נוספים המאפיינים (צבעים אמיתיים וכן הלאה). לפרטים נוספים על אחסון נתוני מסך, ראה SAVE SCREENS.

## פורמט

LOAD string-expr SCREENS

## LPRINT הדפסה במדפסת

## מיקום במקלות

EXTEND MODE  
C

## הצהרה/פקודה

LPRINT גורמת למדפסת מסוג סינקלייר להדפיס פריט מידע באותו אופן בו PRINT גורמת לפרוט להופיע על המסך.

**כיצד להשתמש ב־LPRINT**  
LPRINT עשויה ליצור הצהרה בתכנית או פקודה ישירה. היא מלווה בפריט מידע היכולים להיות מופרדים בנקודה-פסיק, פסיקים, או גרשיים, לדוגמה

```
60 LPRINT "Number" : x'
   "Name" : n$ , "Age" : a
```

כאשר הפלט עובר למדפסת, הנתונים מודפסים באותו פורמט כפי ש־PRINT היתה מציגה אותם על המסך. הצהרות או פקודות LPRINT עשויה לכלול גם הצהרות TAB, בקרי CHR\$ מסומים, הצהרות INVERSE וקודי בקרה עם אותה השפעה כמו PRINT

ניתן לכלול גם הצהרת AT, אך מתעלמים ממספר השורה ופריט המידע מודפס בנקודת העמודה הנתונה באותה השורה.

## פורמט

```
LPRINT [TAB int-num-expr,]
[AT int-num-expr,int-num-expr,]
[CHR$(int-num-expr),][statement,]
[num-expr][string-expr[,].[,]]
```

## MERGE

## מיקום במקלות

EXTEND MODE  
SYMBOL SHIFT T

## הצהרה/פקודה

MERGE מאפשרת מיזוג של שתי תכניות.

**כיצד להשתמש ב־MERGE**  
ניתן להשתמש ב־MERGE ליצירת הצהרה בתכנית או כפקודה ישירה. היא מלווה בשם קובץ בצורת ערך מחרוזת, לדוגמה

## 500 MERGE "prog 2"

שם הקובץ הבא אחרי MERGE הוא שם התכנית שתתמזג עם התכנית הקיימת בזכרון. שם זה כפוף לאותן מגבלות שם תכנית הקיימות בשימוש עם LOAD MERGE טוענת את התכנית החדשה בלי למחוק תחילה את התכנית הקיימת. ברם, התכנית החדשה נכתבת על גבי התכנית הקיימת במקומות בהם קיימים אותם מספרי שורות ושמות משתנים כמו בתכנית הקיימת.

## פורמט

MERGE string-expr

## MOVE

פקודת טיפול בקבצים בכונן זעיר. ראה מדרוך כונן זעיר ומימשק 1.

## NEW

## מיקום במקלות

A

## הצהרה/פקודה

NEW מנקה את מרחב זכרון מערכת בייסיק (כל התחום עד RAMTOP) ומוחקת כל תכנית הקיימת בחלק זה של הזכרון.

**כיצד להשתמש ב־NEW**  
NEW משמשת לרוב כפקודה ישירה אך עשויה גם ליצור הצהרה בתכנית. היא פועלת בנפרד, בעת הביצוע, נמחקים התכנית והמשתנים. הזכרון נמחק עד למחיצת RAMTOP כך שתווי גרפיקה בהגדרת משתמש, אשר מאוחסנים מעל RAMTOP, אינם מושפעים.

## פורמט

NEW

## NEXT

## מיקום במקלות

## הצהרה/פקודה

NEXT נמצאת תמיד בשימוש יחד עם FOR ליצירת לולאת FOR NEXT

## כיצד להשתמש ב־NEXT

NEXT משמשת לרוב ליצירת הצהרה בתכנית להשלמת לולאת FOR NEXT. היא מלווה באות שהיא משתנה הבקרה בלולאה, לדוגמה

## 90 NEXT A

בבייסיק של סינקלייר, חייבים לכלול את משתנה הבקרה. ראה FOR לפרטים נוספים על לולאות FOR NEXT.

## פורמט

NEXT letter

## NOT

## מיקום במקלות

SYMBOL SHIFT S

## אופרטור לוגי/פונקציה

NOT משמש להיפוך מצב אמת של תנאי כך שתנאי שקרי הופך לאמת ולהיפך.

## כיצד להשתמש ב־NOT

NOT מלווה בתנאי או בערך מספרי, לדוגמה

```
90 IF NOT x = y + z THEN PRINT
   "Wrong"
```

```
90 LET correct = x = y + z : IF NOT
   correct THEN PRINT "Wrong"
```



מפעילה את שקע MIC וסיבית 4 את הרמקול. כתובת שער 251 מפעילה את המדפסת, ושערים 247, 254 ו-239 משמשים עבור רכיבים היקפיים אחרים.

#### פורמט

OUT int-num-expr, int num-expr

## OVER

#### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT N

#### הצהרה/פקודה

OVER משמשת להדפסת תו אחד על גבי תו אחר. ניתן גם להתוות בעזרתה נקודות או לצייר קווים ועקומות בצבע נייר במקום בצבע דיו.

כיצד להשתמש ב-OVER  
OVER משמשת בר"כ ליצירת הצהרה בתכנית. היא מלווה בערך מספר, לדוגמה

80 OVER 1

הערך הבא אחרי OVER מעוגל לשלם הקרוב ביותר ועשוי להיות איפוא 0 או 1. OVER 0, שהוא ברירת המחדל (קבוע מראש), גורמת לכל תו למחוק תו קודם שהיה באותו מיקום ולהחליפו. OVER 1 גורמת לשילוב של כל שני תוים המוצגים באותו מיקום תו.

OVER ניתנת להטמעה (הכנסה) בתוך הצהרת PRINT או INPUT באותו אופן כמו IN כן שהיא משפיעה רק על תוים המוצגים על ההצהרה. ההצהרה הבאה, למשל, מדגישה מילה בקו

60 PRINT AT 11, 15; "YES";  
OVER 1; AT 11, 15; "\_\_\_"

ברם, שים לב שהתוים משולבים כך שבמקום בו צבעי הדיו חופפים, ניתן צבע הנייר.

OVER במצב הסדרה גבוהה OVER ניתן לשימוש עם PLOT, CIRCLE, DRAW, ללא OVER, קווים ועקומות יכולים לחפוף זה את זה, אך הם חייבים להיות באותו צבע דיו אחרת צבע הדיו בכל מיקום התו ישתנה במקומות החציית. אם מופעלת OVER 1, קווים או עקומות יוצרים את צבע הנייר במקומות החפיפה או פגישת תוים אחרים.

#### OR בתור פונקציה

הספקטרום  $2x + 2x$  מקצה ערך מספרי של 1 לתנאי אמת רש לתנאי שקר. הוא מכיר כל ערך שונה מאפס כאמת רש כשקר. ניתן איפוא להקדים או ללוות את OR בערך מספרי, לדוגמה

40 LET  $x = y$  OR  $z$

המשתנה  $x$  מקבל ערך של 1 אם  $z$  שונה מאפס או תנאי אמת, או ערך של  $y$  אם  $z$  שווה 0 או תנאי שקר. הדבר שימושי באריתמטיקה. בדוגמה הבאה, מחיר הכרטיס (fare) מחולק ב-2 אם הגיל (age) קטן מ-14.

60 PRINT fare \* (0.5 OR age > 13)

אם הגיל קטן מ-14, התנאי  $age > 13$  הוא שקר, ולכן מחיר הכרטיס מוכפל ב-0.5. אם  $age > 13$  נכון, אזי המחיר מוכפל ב-1.

שים לב שהספקטרום אינו מחשב צירופים של ערכים מספריים בהתאם לטבלאות אמת תקינות.

#### פורמט

cond OR cond  
num-expr OR num-expr

## OUT

#### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT O

#### הצהרה/פקודה

OUT שולחת בית לכתיבת שער (Port) נתונה במטרה להניע התקן פלט.

כיצד להשתמש ב-OUT  
ניתן להשתמש ב-OUT ליצירת הצהרה בתכנית או כפקודה ישירה. היא מלווה בשני ערכים מספריים, המופרדים בפסיק, לדוגמה

40 OUT 254, 3

שני הערכים מעוגלים לשלם הקרוב ביותר. הערך הראשון (254 לעיל) עשוי להיות בתחום בין 0 ל-65535 והוא כתובת השער. הערך השני (3) עשוי להמצא בין 0 ל-255 והוא הבית העתידי להישלח לאותה כתובת. סיביות 0 עד 2 של בית הפלט לכתובת השער 254 קובעות את צבע הגבול; הדוגמה שלעיל מחזירה איפוא צבע גבול סגול. סיבית 3 בכתובת זו

כאשר NOT מלווה בתנאי  $x = y + z$  לעיל, הספקטרום מקצה ראשית ערך של 1 לתנאי אם הוא אמת רש אם הוא שקר. NOT משמש, איפוא, כפונקציה להיפוך הערך שנוצר, כך שניתן לבדוק את התנאי ההפכי. שים לב שתנאי חייב להיות בתוך סוגריים אם הוא מכיל AND או OR. אם NOT מלווה בערך מספרי, היא מחזירה 0 אם הערך המלווה שונה מאפס ו-1 אם הערך המלווה הוא 0. לכן בדוגמאות שלעיל, הספקטרום מדפיס "Wrong" אם  $x < y + z$  או אם correct הוא בעל ערך 0.

#### פורמט

NOT cond  
NOT num-expr

## OPEN #

פקודת טיפול בקבצים בכונן זעיר. ראה מדרוך כונן זעיר ומימשק 1.

## OR

#### מיקום במקלדת

SYMBOL SHIFT U

#### אופרטור לוגי/פונקציה

OR פועלת כאופרטור לוגי לבדיקת מצב אמת של צירוף תנאים. אם תנאי אחד או יותר הם אמת, אזי כל הצירוף הוא אמת. OR פועלת גם כפונקציה לבצוע פעולות בינאריות על שני ערכים מספריים.

כיצד להשתמש ב-OR  
בתור אופרטור לוגי, OR מקשרת בין שני תנאים בהצהרה בה יש לבדוק את אמיתות כל הצירוף, לדוגמה

70 IF INKEY\$ = "N" OR  
INKEY\$ = "n" THEN STOP

אם אחד או שני התנאים הם אמת, אזי הצירוף כולו אמת. בשורה שלעיל, אחד מהתנאים ("N" = INKEY\$ או "n" = INKEY\$) הופך לאמת ברגע שמקש N נלחץ, ללא קשר למצב מקשי CAPS SHIFT או CAPS LOCK כל הצירוף הופך לאמת והתכנית נעצרת.



התוויית נקודות או ציור קווים ועקומות שנית באותו מקום כאשר מופעל 1 OVER גורמת להעלמותם מהמסך.

#### פורמט

OVER int-num-expr

## PAPER

#### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT C

#### הצהרה/פקודה

PAPER משמשת לבחירת צבע רנייר או הרקע עבור תצוגת המסך. זה עשוי להיות צבע הרקע על פני כל שטח התצוגה, או הצבע מאחורי תוים, נקודות, או קווים בודדים המופיעים במיקומי תוים בודדים.

#### כיצד להשתמש ב-PAPER

ניתן להשתמש ב-PAPER ליצירת הצהרה בתכנית או כפקודה ישירה. הוא מלווה בערך מספרי, לדוגמה

80 PAPER x

הערך המלווה את PAPER מעוגל לשלם הקרוב ביותר ועשוי לנוע בין 0 עד 9. צבעי הנייר הניתנים זהים לאלה הניתנים ע"י INK. צבעי נייר עשויים גם להיות מקיפים או מקומיים ע"י הטמעתם (הכנסתם) בהצהרות תצוגה בדיוק כמו עם צבעי דיו. ראה INK לפרטים נוספים.

כאשר תוים מודפסים לאחר הצהרת PAPER, מקיפה או מקומית, הרקע סביב כל מיקום התו שעליו פעלו משתנה לצבע הנבחר. הדבר אמור גם לגבי מצב בו הותו נקודות או ציורי קווים ועקומות בעזרת הצהרת PAPER מוטמעת, אך לא אחרי פקודה או הצהרה מקיפות. כדי ליצור רקע צבעוני על כל שטח התצוגה, הכרחי להשתמש ב-CLS לאחר הצהרת PAPER. כל התצוגה עוברת לצבע זה, אשר נשאר גם אחרי צבע הרקע הכללי.

#### פורמט

PAPER int num-expr [ : ]

## PAUSE

#### מיקום במקלדת

M

#### הצהרה/פקודה

ניתן להשתמש ב-PAUSE לצורך השהיית תכנית לפרק זמן מוגדר או בלתי מוגבל.

#### כיצד להשתמש ב-PAUSE

PAUSE משמשת בד"כ ליצירת הצהרה בתכנית. היא מלווה בערך מספרי, לדוגמה

130 PAUSE 100

הערך המלווה את PAUSE מעוגל לשלם הקרוב ביותר ועשוי להמצא בתחום בין 0 ל-65535. הערך מגדיר את ההשהייה כמספר מסגרות תמונת הטלביזיה, כך שערך של 50 יוצר השהייה בת 1 שנייה בישראל, בריטניה ואירופה בהן תדר המסגרת הוא 50 הרץ. ברם, שים לב שניתן להפסיק כל השהייה ע"י לחיצת מקש כלשהו. PAUSE נותנת השהייה בלתי מוגבלת הנמשכת עד ללחיצת מקש כלשהו.

#### פורמט

PAUSE int-num-expr

## PEEK

#### מיקום במקלדת

EXTEND MODE  
O

#### פונקציה

PEEK נותנת את ערך הבית המאוחסן בכתובת מסוימת בזכרון.

כיצד להשתמש ב-PEEK  
PEEK מלווה בערך מספרי, לדוגמה

80 LET x = PEEK(256 \* y)

שים לב שבטוי חייב להיות בתוך סוגריים. הערך המלווה את PEEK מעוגל לשלם הקרוב ביותר אם נדרש, ויכול לנוע בין 0 עד 65535 עבור כתובת בזכרון. PEEK מחזירה את ערך הבית (מספר בין 0 ל-255) בכתובת הנתונה.

#### דוגמה

מספר מסגרות תצוגת הטלביזיה שהתרחשו מאז הופעל הספקטרום

לאחרונה מאוחסן בכתובות 23672 עד 23674. הצצה לכתובות אלה, תוך כדי יצירת המסגרות בקצב קבוע נותנת שיטה למדידת זמן. השורה הבאה מציגה את הזמן בשניות שעבר מאז הופעל הספקטרום לאחרונה (פחות הזמן שנוצל להפקת קול והפעלת רכיבים היקפיים כרשמקול ומדפסת).

```
10 PRINT (PEEK 23672 + 256 *  
PEEK 23673 + 65536 * PEEK  
23674)/50
```

הערה - אם תדר הרשת הוא 60 הרץ ולא 50 הרץ (תקן בריטי), שנה 50 ל-60.

#### פורמט

PEEK int-num-const  
PEEK int-num-var  
PEEK (int-num-expr)

## PI

#### מיקום במקלדת

EXTEND MODE  
M

#### פונקציה

PI נותנת את הערך של פי (3.14159265...) לשימוש בחישובים. PI הוא היחס בין היקף העיגול לקוטרו.

#### כיצד להשתמש ב-PI

PI אינה דורשת כל ערכים או משתנים כאשר היא מופעלת בהצהרה או פקודה, לדוגמה

DRAW 255,0,-PI

PI מחזירה ערך של 3.14159265... שהפקודה שלעיל מציירת חצי מעגל גדול על המסך.

#### פורמט

PI

## PLOT

#### מיקום במקלדת

Q

#### הצהרה/פקודה

PLOT משמשת בגרפיקה בהפרדה גבוהה להתוויית פיקסל או נקודת צבע במיקום מסוים על המסך.

#### כיצד להשתמש ב-PLOT

PLOT משמשת ליצירת הצהרה בתכנית או כפקודה. בד"כ היא מלווה



## PRINT

### מיקום במקלדת

P

### הצהרה/פקודה

PRINT מציגה מידע על המסך. חמידע עשוי להיות כל תו בודד או רצף תווים. הצהרת PRINT עשויה לכלול מילות מפתח אחרות להגדרת מיקום וצבע המידע.

### כיצד להשתמש ב-PRINT

ניתן להשתמש ב-PRINT לחוד או כשהיא מלווה במידע. מידע זה עשוי להיות בצורת כל בטוי מספרי או מחרוזת, או תערובת שלהם. כאשר משתמשים ב-PRINT עם נתונים, שני פריטים נפרדים או יותר חייבים להיות מופרדים בנקודה-פסיק, פסיק או גרש.

ניתן להכניס מילות מפתח מסוימות אחרות בכל סדר שהוא בין PRINT והנתונים, בתנאי שכל הצהרה הנוצרת ע"י מילות המפתח מסתיימת בנקודה-פסיק. מילות מפתח אלה הן: FLASH, PAPER, INK, AT, TAB, CHR\$, OVER, INVERSE, BRIGHT.

### PRINT עם מחרוזות

PRINT לחוד או בלוויית מחרוזת אפס ("") מציגה שורה ריקה ומעבירה את הסמן לתחילת השורה הבאה. PRINT מלווה בקבוע מחרוזת (תווים) כלשהם בין מרכאות כפולות) מציגה את התווים כפי שהם מופיעים בין סימני המרכאות. הפקודה

```
PRINT "3/542/76/21"
```

לדוגמה, מציגה

```
3/542/76/21
```

PRINT מלווה במשתנה מחרוזת או בטוי מציגה את המחרוזת או מחרוזת שהם מייצגים.

### PRINT עם מספרים

PRINT מלווה בבטוי מספרי כלשהו מציגה את ערך הבטוי. מספרים מוצגים בפורמט עשרוני עם עד ל-8 ספרות משמעותיות וללא אפסים עוקבים אחרי הנקודה העשרונית. מספרים גדולים מאד וקטנים מאד מוצגים בפורמט מדעי מקוצר, כשני מספרים המופרדים באות E, זה מצוין

### כיצד להשתמש ב-POINT

POINT מלווה בשני ערכים מספריים המופרדים בפסיק ונמצאים בתוך סוגריים, לדוגמה

```
240 IF POINT (x, y) = 1 THEN  
GOSUB 600
```

שני הערכים המלווים את POINT מעוגלים לשלמים במידת הצורך. הערך הראשון עשוי לנוע בין 0 ל-255 ומגדיר את הקואורדינטה האפקית של פיקסל על המסך. הערך השני עשוי לנוע בין 0 ל-175 ומגדיר קואורדינטה אנכית. POINT מחזירה 1 אם הפיקסל בנקודה המוגדרת הוא צבע דיו או 0 אם הוא צבע נייר.

### פורמט

```
POINT (int-num-expr, int-num-expr)
```

## POKE

### מיקום במקלדת

0

### הצהרה/פקודה

POKE משמשת לשינוי ערך הבית בכתובת מסוימת בזכרון. ערכים בד"כ "נרחפים" למקומות בזכרון במטרה ליצור פעולות שאינן ניתנות לבצוע ע"י מילות מפתח בביוסיק.

### כיצד להשתמש ב-POKE

POKE משמשת ליצירת הצהרה בתכנית או פקודה. היא מלווה בשני ערכים מספריים המופרדים בפסיק, לדוגמה

```
POKE 23609, 255
```

שני הערכים המלווים את POKE מעוגלים לשלמים הקרובים ביותר במידת הצורך. הערך הראשון עשוי לנוע בין 16384 עד 65535 ומהווה כתובת בזכרון ראש. הערך השני עשוי לנוע בין 0 עד 255 ומהווה את הבית האמור להכתב בכתובת המוגדרת. בדוגמה שלעיל, 255 "נרחף" לכתובת 23609, אשר מבקרת את הקול כאשר נלחץ מקש. ערך של 255 יוצר צפצוף ארוך במקום ה"קליק" הרגיל, וערכים אחרים יוצרים צפצוף קצר יותר.

### פורמט

```
POKE int-num-expr,  
int-num-expr
```

בשני ערכים מספריים המופרדים בפסיק, לדוגמה

```
50 PLOT 128, 87
```

שני הערכים המלווים את PLOT מעוגלים לשלמים במידת הצורך. הערך הראשון עשוי לנוע בין 0 ל-255 ומגדיר את הקואורדינטה האפקית של הנקודה על המסך. הערך השני עשוי לנוע בין 0 ל-175 ומגדיר קואורדינטה אנכית. הפיקסל יתווה בד"כ בצבע הדיו הקיים בנקודה המוגדרת – בדוגמה שלעיל, במרכז המסך. שים לב להשפעות הבאות של הצהרות או פקודות צבע על PLOT אחרי 1 OVER, כל נקודה קיימת באותו מיקום משנה את צבעה לצבע הנייר. לאחר 1 INVERSE, הנקודה מותווה בצבע הנייר הקיים. לאחר 1 BRIGHT או 1 FLASH, כל מיקום התו על המסך בהפרדה נמוכה שעליו מותווה הפיקסל יהיה בהיר או מהבהב.

ארבע מילות מפתח אלה: INK ניתנות גם להטמעה בתוך הצהרת PLOT באותו אופן כמו עם PRINT, לדוגמה

```
160 PLOT INK 2: x, y
```

השפעתן זהה אך היא מקומית ומוגבלת לפיקסל המותווה ע"י ההצהרה. אם PAPER מוטמעת בהצהרת PLOT, צבע הנייר של כל מיקום התו סביב הפיקסל משתנה לצבע הנתון.

שים לב ש-PLOT גם מגדירה את נקודת ההתחלה של הצהרת DRAW הבאה.

### פורמט

```
PLOT (statement : )  
int-num-expr, int-num-expr
```

## POINT

### מיקום במקלדת

```
EXTEND MODE  
SYMBOL SHIFT 8
```

### פונקציה

POINT משמשת לבירור האם הצבע בנקודה מסוימת על המסך בהפרדה גבוהה הוא צבע דיו או צבע נייר. POINT אינה בודקת את הצבע עצמו.



מספר שבו החלק הראשון (המנטיסה) מועלית בחזקת החלק השני (המעריך). הפקודה

PRINT 3/542/76/21

לדוגמה, מציגה

3 4680798 E -6

שני פורמט PRINT בעזרת סימני פיסוק

PRINT מלווה בפריטי מידע המופרדים בנקודה-פסיק מציגה את הפריטים כשהם זה בצד זה ללא רווח. הפקודה

PRINT 1:2:3

מציגה

123

PRINT מלווה בפריטי מידע המופרדים בפסיק מציגה כל פריט בתחילת או באמצע השורה בהתאם למיקום הפריט הראשון. הפקודה

PRINT 1,2,3

מציגה

1 2  
3

PRINT מלווה בפריטי מידע המופרדים בגרש מציגה את הפריט שאחרי הגרש בתחילת השורה הבאה. הפקודה

PRINT 1 '2 '3

מציגה

1  
2  
3

אם הצהרת או פקודת PRINT מסתיימת בנקודה-פסיק, פסיק או גרש, אזי הפריט שיוצג על ידי הצהרת PRINT הבאה יושפע באותו אופן.

PRINT ומילות מפתח אחרות ניתן ללוות את PRINT ב-TAB, ערך מספרי, נקודה-פסיק ואחרים פריט מידע, לדוגמה

60 PRINT TAB x: a\$

הערך הבא אחרי TAB (x שלעיל) מעוגל לשלם הקרוב ביותר במידת הצורך ואז מחולק ב-32 והשארי מוחזרת בצורת ערך הנע בין 0 ל-31. פריט המידע מוצג על המסך במספר עמודה זה באותה השורה או בשורה הבאה.

ניתן ללוות את PRINT ב-AT ושני ערכים מספריים המופרדים בפסיק, נקודה-פסיק ופריט מידע, לדוגמה

50 PRINT AT 1, c: "Data"

הערך הראשון (ו שלעיל) עשוי לנוע בין 0 ל-21 ומגדיר את מספר השורה בה יוצג המידע. הערך השני (c) עשוי לנוע בין 0 ל-31 ומגדיר את מספר העמודה בה יוצג התו הראשון או הספרה הראשונה של פריט המידע. ערכים לא שלמים מתקבלים ומעוגלים לשלם הקרוב ביותר, הפקודה PRINT AT 11, 16: " " מציגה כוכב במרכז המסך. ניתן גם ללוות את PRINT באחת או יותר פונקציות CHR\$. ראה CHR\$ לפריטים נוספים.

PRINT ומילות מפתח לצבע התצוגה המופקת על ידי PRINT מושפעת מהצהרות או פקודות צבע הניתנות על ידי BRIGHT, FLASH, PAPER, INK, INVERSE OVER. הנמצאות אותו זמן בפעולה. ניתן גם ללוות את PRINT באחת או יותר משש הצהרות אלה כשכל אחת מלווה בנקודה-פסיק לפני פריט המידע, לדוגמה

50 PRINT AT 11, 16: INK 2:  
FLASH 1: " " "

פריט המידע מוצג עם המאפיינים המוגדרים על ידי מילות המפתח לצבע. מאפיינים אלה הם מקומיים ושימיים רק לפריט המוצג לאחר בצוע הצהרת PRINT, הם חוזרים לערכיהם בברירת המחדל או ערכים מקיפים שהוכרו מקודם. PRINT כפופה גם לקודי בקרת צבע מקומיים המוכנסים עם המידע (ראה עמ' 33).

פורמט

PRINT [TAB int-num-expr, ]  
[AT int-num-expr, int-num-expr: ]  
[CHR\$ (int-num-expr); ][statement: ]  
[num-expr][string-expr][. ][. ][ ]

## RANDOMIZE

מיקום במקלות

T

הצהרה/פקודה

RANDOMIZE, המופיעה על המקלות כ-RND, משמשת בהקשר עם RND ליצירת רצפי מספרים שהם אקראיים או צפויים מראש.

כיצד להשתמש ב-RANDOMIZE  
RANDOMIZE משמשת ליצירת הצהרה בתכנית או כפקודה. ניתן ללוותה, כאופציה, בערך מספרי, לדוגמה

RANDOMIZE 1  
10 RANDOMIZE

הערך המלווה את RANDOMIZE מעוגל לשלם הקרוב ביותר במידת הצורך ועשוי לנוע בין 0 עד 65535. ערך גדול מ-65535 קובע את משתנה המערכת SEED לערך זה, הבא אחרי רצף המספרים הנוצר תמיד על ידי RND (ראה עמ' 48 למידע על משתני מערכת). הרצף המעשי לא תלוי בערך של RANDOMIZE. RANDOMIZE ON מלווה ב-0 או ב-1, ללא ערך, אזי SEED מקבל ערך של משתנה מערכת אחר הקרוי FRAMES, אשר מונה את המסגרות המוצגות על מסך הטלביזיה מאז הופעל הספקטרום לאחרונה. מכיון ש-SEED משתנה 50 או 60 פעם בשנייה, רצף המספרים הנוצר על ידי RND לאחר RANDOMIZE או 0 RANDOMIZE הוא אקראי ביותר. RANDOMIZE אינו בשימוש, RND יוצר את אותו רצף מספרים מזמן הפעלת המחשב ואחרי שימוש בכפתור RESET או פקודת NEW.

פורמט

RANDOMIZE [int-num-expr]

## READ

מיקום במקלות

EXTEND MODE  
A

הצהרה/פקודה

READ משמשת בהקשר עם DATA להקצאת ערכים עבור משתנים תוך שימוש בערכים שבהצהרת DATA.

כיצד להשתמש ב-READ  
READ משמשת בד"כ ליצירת הצהרה בתכנית. היא מלווה באחד או יותר



## פונקציה

RND משמשת ליצירת מספר אקראי.

**כיצד להשתמש ב-RND**  
RND מופיעה לחוד בהצהרה או פקודה, לדוגמה

```
60 LET X = RND
```

RND מחזירה מספר אקראי קטן מ-1 וגדול או שווה ל-0.

כאשר הספקטרום מופעל או מאותחל, או שהוקש NEW, יוחזרו מספרים ע"י RND באותו סדר. הרצף נוצר ע"י שימוש בחזקות של 75 (75, 75\*75, 75\*75\*75 וכן הלאה), חלוקת כל חזקה ב-65537 וניצול השארית בלבד ע"י החסרת 1 ממנה וחלוקת התוצאה ב-65536.

אם נדרש רצף אקראי יותר או רצף קבוע אחר, השתמש ב-RANDOMIZE לפני RND

**מספרים שלמים אקראיים**  
הצהרות ופקודות רבות בספקטרום, כמו INK ו-CHR\$, מעגלות מספרים לשלם הקרוב ביותר וניתן להשתמש בהם ישירות עם RND. לדוגמה,  $7 * \text{RND} \text{ INK}$  יוצרת צבע דיו אקראי, כדי לקבל מספר שלם אקראי מ-1 עד  $x$  ניתן להשתמש ב- $\text{INT}(\text{RND} * x) + 1$  ליצירת מספר שלם אקראי מ-0 עד  $x$ , השתמש ב- $\text{INT}(\text{RND} * x + 0.5)$

פורמט

RND

## RUN

מיקום במקלות

R

הצהרה/פקודה

RUN גורמת לתכנית לרוץ, בד"כ מהשורה הראשונה.

**כיצד להשתמש ב-RUN**  
ניתן להשתמש ב-RUN כפקודה ישירה או ליצירת הצהרה בתכנית. היא מלווה, לפי בחירה, בערך מספרי, לדוגמה

```
RUN 50
```

אם אין עודך אחרי RUN, התכנית רצה מהשורה הראשונה. אם נכלל ערך מסוים, הוא מעוגל לשלם הקרוב ביותר במידת הצורך והתכנית רצה

## כיצד להשתמש ב-RESTORE

RESTORE יוצרת בד"כ הצהרה בתכנית. היא מלווה לפי בחירה בערך מספרי, לדוגמה

```
160 RESTORE 800
```

הערך הבא אחרי RESTORE מעוגל לשלם הקרוב ביותר במידת הצורך, ואז עליו להיות מספר שורה בתכנית אשר מכילה הצהרת DATA. לאחר RESTORE, הצהרת READ הבאה תקצה את ערכי הצהרת DATA הנ"ל. אם השורה הממוספרת אינה קיימת או שאינה מכילה הצהרת DATA, אזי READ תפנה להצהרת DATA הבאה אחרי שורה זו.

אם RESTORE מלווה ב-0 או ללא עודך מלווה, אזי פקודת READ הבאה תמנה להצהרת DATA הראשונה בתכנית.

פורמט

```
RESTORE [int-num-expr]
```

## RETURN

מיקום במקלות

Y

הצהרה/פקודה

RETURN משמשת לסיום שיגרה והחזרת המחשב לתכנית הראשית או שיגרה קודמת.

**כיצד להשתמש ב-RETURN**  
RETURN משמשת בד"כ ליצירת הצהרה בתכנית. היא מופיעה לחוד בסיום השגרה, לדוגמה

```
1000 RETURN
```

בעת הבצוע, התכנית מסתעפת להצהרה הבאה אחרי הצהרת GOSUB האחרונה שבוצעה. ראה GOSUB לפרטים נוספים.

פורמט

RETURN

## RND מספר אקראי

מיקום במקלות

```
EXTEND MODE  
T
```

ערכים מספריים או משתני מחזרת המופרדים בפסיקים, לדוגמה

```
20 READ A$, X
```

כאשר READ מתבצעת בראשונה, היא לוקחת את אותו מספר ערכים כמספר המשתנים מתחילת רשימת DATA הראשונה ומקצה לפי הסדר את הערכים למשתנים. כאשר READ מתבצעת בשנית, המערך הבא של ערכי DATA מוקצה למשתנים הנקראים בהצהרת READ וכן הלאה. לפרטים נוספים, ראה DATA

פורמט

```
READ num-var [, num-var]  
[ , string-var]  
READ string-var [, num-var]  
[ , string-var]
```

## REM הערה

מיקום במקלות

E

הצהרה

REM משמשת לכתיבת הערות או תזכורות בתכנית. אלה עשויות להיות הכותרת ושם יוצר התכנית, והסברים לשורות בתכנית כמו מטרת משתנה מסוים. ההערות לא משתתפות בריצת התכנית וניתן לראותן רק ברישום.

**כיצד להשתמש ב-REM**

REM יוצרת שורה משל עצמה בתכנית או מהווה הצהרה אחרונה בשורה. היא מלווה בהערה כלשהי הניתנת להקשה לפי הצורך, לדוגמה

```
80 INPUT N$: REM N$ is name
```

כאשר המחשב נתקל ב-REM, הוא מתעלם מכל דבר הבא אחרי REM באותה שורה.

פורמט

```
REM כלשהם
```

## RESTORE

מיקום במקלות

```
EXTEND MODE  
S
```

הצהרה/פקודה

RESTORE משמשת בהקשר עם READ כדי ש-READ תיקח ערכים מהצהרת DATA מסוימת במקום הצהרת DATA הראשונה או שנייה בתכנית.



**הצהרה/פקודה**

SAVE DATA מאחסנת מערך על סרט.  
ניתן לטעון חזרה את המערך בעזרת  
LOAD DATA

**כיצד להשתמש ב־SAVE DATA**  
ניתן להשתמש ב־SAVE DATA ליצירת  
הצהרה בתכנית או כפקודה ישירה.  
SAVE מלווה בשם קובץ, אחריו  
DATA, אות או אות וסימן \$, ולבסוף  
צמד סוגריים ריקים, לדוגמה

```
450 SAVE "numbers" DATA n( )
750 SAVE "names" DATA n$( )
```

שם קובץ המערך עשוי להכיל עד 10  
תווים. האות או אות \$ שאחרי DATA  
היא שם המערך בתכנית המיועד  
לאחסנה על סרט. המערך נשלח  
לסרט באותה צורה כפי שתכנית  
נשלחת לסרט ע"י SAVE

**פורמט**

SAVE string-expr DATA letter [ \$ ] ( )

**SAVE SCREEN\$****מיקום במקלדת**

S  
EXTEND MODE  
SYMBOL SHIFT K

**פקודה/הצהרה**

SAVE SCREEN\$ מאחסנת את תצוגת  
המסך על סרט. ניתן לטעון אותה  
חזרה למחשב מאוחר יותר בעזרת  
LOAD SCREEN\$

**כיצד להשתמש ב־SAVE SCREEN\$**  
ניתן להשתמש ב־SAVE SCREEN\$  
כפקודה ישירה או ליצירת הצהרה  
בתכנית. SAVE מלווה בשם קובץ  
שהוא משתנה מחרוזת ואחריו  
SAVE SCREEN\$, לדוגמה

```
SAVE "picture" SCREEN$
```

שם הקובץ עשוי להכיל עד 10 תווים.  
התצוגה נשלחת לסרט באותו אופן  
כפי שתכנית נשלחת לסרט ע"י SAVE

**פורמט**

SAVE string-expr SCREEN\$

**SCREEN\$****מיקום במקלדת**

EXTEND MODE  
SYMBOL SHIFT K

**פונקציה**

SCREEN\$ מגלה איזה תו מופיע

המספר המוגדר, או אם לא קיימת  
שורה כזאת, מהשורה הבאה בתכנית.  
למעשה, השימוש ב־LINE גורם לכל  
התכנית לרוץ אוטומטית.

**פורמט**

SAVE string-expr [LINE int-num-expr]

**SAVE CODE****מיקום במקלדת**

S  
EXTEND MODE  
I

**פקודה/הצהרה**

SAVE CODE שולחת קטע מידע  
בזכרון לרשמקול קלטות כדי להקליטו  
על סרט. ניתן תמיד להחזיר את  
המידע לזכרון בעזרת LOAD CODE

**כיצד להשתמש ב־SAVE CODE**  
ניתן להשתמש ב־SAVE CODE  
כפקודה ישירה או ליצירת הצהרה  
בתכנית. SAVE מלווה בשם קובץ  
שהוא משתנה מחרוזת ואחריו CODE,  
המלווה בשני ערכים מספריים  
המופיעים בפסיק, לדוגמה

```
SAVE "picture" CODE 16384,6912
```

שם הקובץ המלווה את SAVE עשוי  
להכיל עד 10 תווים. שני הערכים  
הבאים אחרי CODE מעוגלים במידת  
הצורך לשלמים הקרובים ביותר. הערך  
הראשון נותן את כתובת ההתחלה  
(16384 לעיל) של המידע בזכרון,  
והערך השני (6912) נותן את מספר  
הבתים שיש לאחסן. המידע נשלח  
לסרט באותה צורה כפי שתכנית  
נשלחת לסרט ע"י SAVE  
המידע הנשמר בפקודה שלעיל הוא  
תצוגת המסך.

**פורמט**

SAVE string-expr CODE  
int-num-expr, int-num-expr

**SAVE DATA****מיקום במקלדת**

S  
EXTEND MODE  
D

משורה זאת. אם השורה לא קיימת,  
התכנית תרוץ מהשורה הבאה  
בתכנית. שים לב ש־RUN מבצעת  
CLEAR לפני הרצת התכנית, כך  
שערכי המשתנים נמחקים. כדי למונע  
זאת, השתמש ב־GOTO מלווה במספר  
שורה.

אם תכנית נשמרה באמצעות LINE,  
היא תרוץ אוטומטית לאחר הטעינה  
ולא נדרש שימוש ב־RUN

**פורמט**

RUN [int-num-expr]

**SAVE****מיקום במקלדת**

S

**פקודה/הצהרה**

SAVE שולחת תכנית לרשמקול  
במטרה לאחסנה על סרט.

**כיצד להשתמש ב־SAVE**  
משמשת בז' כפקודה ישירה  
אך עשויה גם ליצור הצהרה בתכנית.  
היא מלווה בשם קובץ שהוא משתנה  
מחרוזת, לדוגמה

```
SAVE "filename"
```

שם הקובץ עשוי להכיל עד 10 תווים.  
בעת הבצוע, ההודעה

Start tape, then press any key

מוצגת על המסך. בלחיצת מקש  
כלשהו, התכנית נשלחת לרשמקול  
קלטות, ולבסוף מופיע הדווח  
OK, 0, 1.

שים לב ש־SAVE פועלת בצורה  
שונה כאשר מחובר כונן זעיר. דאה  
פרטים במדריך כונן זעיר ומימשק 1.

**רצה אוטומטית**

אם יש להריץ אוטומטית את התכנית  
בזמן הטעינה, יש להשתמש ב־SAVE  
יחד עם LINE. שם התכנית מלווה ב־  
LINE וערך מספרי, לדוגמה

```
SAVE "filename" LINE 1
```

הערך המלווה את LINE מעוגל לשלם  
הקרוב ביותר במידת הצורך, ואזי עליו  
להיות 1 או מספר שורה בתכנית.  
התכנית נשלחת לסרט באותה צורה  
כמו עם SAVE. בעת הטעינה, התכנית  
רצה באופן אוטומטי מהשורה בעלת



## STOP

### מיקום במקלדת

SYMBOL SHIFT A

### הצהרה/פקודה

STOP עוצרת תכנית בנקודה מסוימת. יתכן ונדרש להשתמש ב־STOP לסיום התכנית הראשית במטרה לבודד את השגרות בקטע נפרד. STOP שימושית מאד גם לצורך ניפוי שגיאות בתכנית.

**כיצד להשתמש ב־STOP**  
STOP משמשת בד"כ ליצירת הצהרה בתכנית. היא מופיעה לחד, לדוגמה

650 STOP

בעת הבצוע, התכנית נעצרת והדווח

9 STOP statement

מופיע עם מספר השורה וההצהרה בה נפסקה התכנית.

נוהלי ניפוי שגיאות, כמו תצוגות ושינוי ערכי משתנים, ניתנים לכיצוע לאחר הקשת STOP. הכנסת CONTINUE לאחר מכן, גורמת לתכנית להמשיך בהצהרה הבאה עם הערכים החדשים.

### פורמט

STOP

## STR\$

### מיקום במקלדת

EXTEND MODE  
Y

### פונקציה

STR\$ הופכת מספר למחרוזת.

**כיצד להשתמש ב־STR\$**  
STR\$ מלווה בערך מספרי, לדוגמה

90 LET A\$ = STR\$ X

בטוי חייב להיות בתוך סוגריים. STR\$ מחזירה את ערך הארגומנט (X לעיל) כקבוע מחרוזת. אם הוקצה ל־X ערך של 65, אזי ההצהרה שלעיל מקצה ל־A\$ ערך של "65".

### פורמט

STR\$ num-const  
STR\$ num-var  
STR\$ (num-expr)

### פונקציה

מה מבצעת SIN?  
SIN נותנת את ערך הסינוס של זווית.

**כיצד להשתמש ב־SIN**  
SIN מלווה בערך מספרי, לדוגמה

80 LET X = SIN Y

בטוי חייב להיות בתוך סוגריים. הערך המלווה את SIN הוא הזווית ברדיאנים, ראו מחזירה את סינוס הזווית. ניתן להפוך מעלות לרדיאנים ע"י הכפלה ב־PI/180

שים לב ש־SIN מחזירה ערך חיובי עבור זוויות בין 0 ל־180 מעלות, וערך שלילי עבור זוויות בין 180 ל־360 מעלות.

### דוגמה

הפקודה

PRINT SIN (30 \* PI/180)

מציגה 0.5, הסינוס של 30 מעלות.

### פורמט

SIN num-const  
SIN num-var  
SIN (num-expr)

## SQR

### שורש רבועי

### מיקום במקלדת

EXTEND MODE  
H

### פונקציה

SQR נותנת את השורש הרבועי של מספר.

**כיצד להשתמש ב־SQR**  
SQR מלווה בערך מספרי, לדוגמה

70 LET X = SQR Y

בטוי חייב להיות בתוך סוגריים. הערך המלווה את SQR (Y לעיל) חייב להיות גדול מאפס, ו־SQR מחזירה את השורש הרבועי שלו.

### פורמט

SQR num-const  
SQR num-var  
SQR (num-expr)

## STEP

### מיקום במקלדת

SYMBOL SHIFT D

ראה FOR

בנקודה מסוימת על המסך.

**כיצד להשתמש ב־SCREEN\$**  
SCREEN\$ מלווה בשני ערכים מספריים המופרדים בפסיק ובתוך סוגריים, לדוגמה

160 IF SCREEN\$ (I,C) = "X"  
THEN PRINT "CRASH"

הערכים המלווים את SCREEN\$ מעוגלים במידת הצורך לשלם הקרוב ביותר.

הערך הראשון (I שלעיל) עשוי לנוע בין 0 ל־21 ונותן את מספר השורה של נקודה על המסך. הערך השני (C לעיל) עשוי לנוע מ־0 עד 31 ונותן את מספר העמודה של הנקודה. SCREEN\$ מחזירה את התו המוצג בנקודה זו כקבוע מחרוזת (התו בתוך סימני מרכאות, "X" לעיל לדוגמה). אם לא קיים כל תו בנקודה זו, SCREEN\$ מחזירה מחרוזת אפס ("). שים לב שניתן גם להשתמש ב־SCREEN\$ עם SAVE ו־LOAD לאחסון תצוגת המסך על סרט ולטעון אותה מסרט, ראה SAVE SCREEN\$ ו־LOAD SCREEN\$ לפרטים.

### פורמט

SCREEN\$ (int-num-expr,int-num-expr)

## SGN

### סימן

### מיקום במקלדת

EXTEND MODE  
F

### פונקציה

SGN מציינת האם מספר הוא חיובי, שלילי או אפס.

**כיצד להשתמש ב־SGN**  
SGN מלווה בערך מספרי, לדוגמה

50 LET X = SGN Y

בטוי חייב להיות בתוך סוגריים. SGN מחזירה 1 אם ערך הארגומנט (Y לעיל) הוא חיובי, -1 אם הוא שלילי ו־0 אם הוא אפס.

### פורמט

SGN num-const  
SGN num-var  
SGN (num-expr)

## SIN

### סינוס

### מיקום במקלדת

EXTEND MODE  
Q



# TAB

מיקום במקלדת

EXTEND MODE

ראה PRINT, LPRINT

# TAN

מיקום במקלדת

EXTEND MODE

פונקציה

TAN נותנת את ערך הטנגנס של זווית.

כיצד להשתמש ב-TAN  
TAN מלווה בערך מספרי, לדוגמה

130 LET x = TAN y

בטוי חייב להיות בתוך סוגריים. הערך המלווה את TAN הוא ערך הזווית ברדיאנים, TAN מחזירה את טנגנס הזווית. ניתן להפוך מעלות לרדיאנים על ידי הכפלה ב-PI/180

שים לב ש-TAN מחזירה ערך חיוכי עבור זוויות בין 0 ל-90 מעלות ובין 180 ל-270 מעלות. עבור זוויות בין 90 ל-180 מעלות וזוויות בין 270 ל-360 מעלות, TAN מחזירה ערך שלילי.

פורמט

TAN num-const  
TAN num-var  
TAN (num expr)

# THEN

מיקום במקלדת

SYMBOL SHIFT G

ראה IF

# TO

מיקום במקלדת

SYMBOL SHIFT F

פונקציה

ל-TO יש שני שימושים עקריים בבסיסיק של סינקלייר. היא משמשת בהקשר עם FOR ליצירת לולאת FOR NEXT (ראה FOR לפרטים) וכמו כן משמשת בפרסת מחרוזת (פיצול מחרוזת לתת-מחרוזות קטנות יותר).

# כיצד להשתמש ב-TO עבור פריסת מחרוזות

TO משמשת להגדרת התו הראשון והתו האחרון של תת-מחרוזת בתוך מחרוזת ראשית. לפני TO מופיע ערך מחרוזת, פתיחת סוגריים, ערך מספרי לבחירה חפשית. היא מלווה בערך מספרי אחר אופציונלי, ולבסוף סגירת הסוגריים. לדוגמה

80 PRINT a\$ (4 TO 7)

בטוי מחרוזת חייב להיות בתוך סוגריים. ערך המחרוזת (a\$ לעיל) הוא המחרוזת המיועדת לפריסה. שני הערכים המספריים (4 ו-7) מגדירים את מקומות התו הראשון והתו האחרון של תת-מחרוזת בתוך המחרוזת. TO מחזירה את תת-המחרוזת (תוים 4 עד 7 של a\$). לערך המספרי הראשון יש ערך ברירת מחדל של 1 ולשני יש ערך מחדל השווה למיקום התו האחרון במחרוזת. ניתן איפוא להשמיט את הערך הראשון אם תת-המחרוזת מתחילה בתו הראשון במחרוזת, וניתן להשמיט את הערך השני אם תת-המחרוזת מסתיימת בתו האחרון במחרוזת.

פורמט

string-const ([num-expr] TO [num expr])  
string-var ([num-expr] TO [num-expr])  
(string expr) ([num-expr] TO [num-expr])

# USR שיגרת משתמש

מיקום במקלדת

EXTEND MODE  
L

פונקציה

USR משמשת לקריאת שגרה בקוד מכונה אשר הוכנסה לזכרון בכתובת מסוימת. כמו כן היא משמשת להכנסת נתונים עבור גרפיקה בהגדרת משתמש במקומות השמורים בראש הזכרון.

USR וקוד מכונה

כדי להשתמש בקוד מכונה, USR מלווה בערך מספרי, לדוגמה

80 PRINT USR 65000  
100 RANDOMIZE USR 65000

בטוי חייב להיות בתוך סוגריים. הערך המלווה את USR מעוגל לשלם הקרוב ביותר והופך לכתובת ההתחלה בזכרון

שבה הוכנסה שגרת קוד מכונה. כל הצהרה המכילה USR קוראת לשגרה בכתובת זו ו-USR מחזירה את ערך התוכן של צמד אוגר bc  
RANDOMIZE USR או RESTORE USR, לדוגמה, מריצות את השגרה בלבד, בעוד PRINT USR מציגה בנוסף את ערך האוגר bc

USR וגרפיקה בהגדרת משתמש כדי ליצור גרפיקה בהגדרת משתמש, משתמשים ב-USR עם USR POKE מלווה בקבוע או משתנה מחרוזת להחזרת כתובת עבור הצהרת POKE, לדוגמה

50 POKE USR "a", 255

ערך המחרוזת הבא אחרי USR עשוי להיות אות בודדת בתחום מ-A עד U או a עד u, כאשר אין הבחנה בין אותיות גדולות וקטנות. USR מחזירה את כתובת ההתחלה של אחד מ-21 חלקי הזכרון השמורים עבור גרפיקה בהגדרת משתמש. כל חלק מכיל 8 כתובות שאליהן מוחדרים בתים ליצירת תו גרפי אחד. ניתן לתת את הבתים בצורה עשרונית או בינארית (ראה BIN).

פורמט

USR int-num-const  
USR int-num-var  
USR (int-num-expr)  
USR string-const  
USR string-var

# VAL ערך

מיקום במקלדת

EXTEND MODE  
J

פונקציה

VAL משנה מחרוזת עם ערך מספרי למספר.

כיצד להשתמש ב-VAL  
VAL מלווה בקבוע או משתנה מחרוזת, לדוגמה

70 LET x = VAL a\$

ערך הקבוע או משתנה המחרוזת מופרד מסימני המרכאות, וחייב לכן להיות ערך מספרי. VAL מחשבת אותו, ומחזירה אותו כקבוע מספרי.



## VERIFY DATA

### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT R  
EXTEND MODE  
□

ראה VERIFY

## VERIFY

### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT R

### פקודה/הצהרה

VERIFY בודקת האם תכנית אוחסנה  
כראוי על סרט לאחר בצוע SAVE

כיצד להשתמש ב־VERIFY  
משתמש ב־VERIFY משמשת בד"כ כפקודה ישירה  
בדיוק כמו LOAD ומלווה בשם  
התכנית, לדוגמה

VERIFY "filename"

כאשר הסרט מופעל, מוצג על המסך  
שם כל תכנית הנמצאת בזכרון וכל  
תכנית על הסרט באותו שם משווה  
עם התכנית בזכרון. אם שתי התכניות  
זהות, הדווח

OK, 1

ניתן על המסך.  
VERIFY מופעלת בצורה שונה כאשר  
מחובר כונן זעיר. ראה פרטים במדריך  
כונן זעיר ומימשק 1.

VERIFY CODE ו־VERIFY DATA  
ניתן להשתמש ב־VERIFY CODE בדיוק  
באותו אופן כמו LOAD CODE לאימות  
אחסון של קטע מידע בזכרון על סרט.  
VERIFY DATA פועלת באותו אופן כמו  
LOAD DATA לבדיקת אחסון תקין של  
מערך על סרט. ראה LOAD CODE ו־  
LOAD DATA לפרטים נוספים.

### פורמט

VERIFY string-expr  
VERIFY string-expr CODE  
[int-num-expr] { , int-num-expr }  
VERIFY string-expr DATA  
letter { \$ } ( )

## VERIFY CODE

### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT R  
EXTEND MODE  
I

ראה VERIFY

### דוגמאות

אם ל־a\$ יש ערך של "435", אזי  
ההצהרה שלעיל מקצה ערך של 435  
ל־x. ברם, VAL יכולה לחשב גם  
בטויים, לדוגמה

10 INPUT a\$ , x  
20 PRINT VAL a\$

ערך המחרוזת המוקצה עבור a\$ צריך  
להיות בטוי המשתמש ב־x. לדוגמה  
"x \* x". ערך מספרי מוקצה עבור x,  
לדוגמה 5. VAL מסירה את סימני  
המרכאות לקבלת x \* x ומחשבת זאת  
תוך שימוש בערך שהוקצה עבור x,  
ומציגה את התוצאה 25.

### פורמט

VAL string-const  
VAL string-var

## VAL\$ (מחרוזת)

### מיקום במקלדת

EXTEND MODE  
SYMBOL SHIFT J

### פונקציה

VAL\$ מחשבת מחרוזת כבטוי  
מחרוזת.

כיצד להשתמש ב־VAL\$  
VAL\$ מלווה במשתנה מחרוזת,  
לדוגמה

130 PRINT VAL\$ a\$

ערך משתנה המחרוזת מופרד מסימני  
המרכאות, וחייב לכן להיות בטוי  
מחרוזת. VAL\$ מחשבת את הבטוי  
ומחזירה את הערך כקבוע מחרוזת.

### דוגמאות

נסה תכנית זאת

10 INPUT a\$ , x\$  
20 PRINT VAL\$ a\$

משתנה המחרוזת המוקצה עבור a\$  
צריך להיות בטוי המשתמש ב־x\$,  
לדוגמה "x\$ + x\$". ערך מחרוזת  
מוקצה עבור x\$, לדוגמה "DO". VAL\$  
מסירה את המרכאות מערך a\$  
לקבלת x\$ + x\$ ומחשבת אותו תוך  
שימוש בערך שהוקצה ל־x\$, ומציגה  
את התוצאה DODO

### פורמט

VAL\$ string-var



# דוחות מסך בספקטרום

כאשר הספקטרום עוצר בצוע של בייסיק, מופיע דוח בתחתית המסך. זה מציין שפקודה או תכנית הסתיימו או שאובחנה שגיאה. כל דוח מורכב ממספר או אות קוד המלווה בהודעה קצרה ואחריה מספרי השורה וההצהרה בה נעצר המחשב. פקודה מתוארת כשורה 0 ובתוך שורה, הצהרה 1 היא בתחילת השורה, הצהרה 2 היא אחרי נקודתיים ראשונות או THEN, וכן הלאה. CONTINUE גורמת בד"כ לתכנית להמשיך מההצהרה המוגדרת בדוח.

## 0 OK

סיום מוצלח, או נסיון לקפוץ למספר שורה הגדול ממה שקיים בתכנית. CONTINUE מתעלמת מדוח זה וממשיכה בהצהרה המוגדרת בדוח הקודם.

## 1 NEXT without FOR

קיימת בתכנית מילת המפתח NEXT ללא FOR בהתאמה וקיים משתנה באותו שם כמו משתנה הבקרה.

## 2 Variable not found

משתנה פשוט היה בשימוש ללא עוד שהוקצה עבורו או עוד שנטען מסרט, או משתנה בקרה שהיה בשימוש עם NEXT ללא קביעתו תחילה בהצהרת FOR, או משתנה מותג שהיה בשימוש לפני הגדרת מימדי המערך בעזרת DIM או טעינת המערך מסרט.

## 3 Subscript wrong

התג נמצא מחוץ למימדי המערך.

## 4 Out of memory

אין מספיק מרחב זכרון להשלמת ההצהרה או הפקודה.

## 5 Out of screen

INPUT יצרה יותר מ-23 שורות לאורך המסך, או שהיה שימוש במספר שורה של 22 ויותר בהצהרת PRINT AT

## 6 Number too big

המחשב ניסה ליצור מספר גדול יותר מ-10<sup>9</sup>.

## 7 RETURN without GOSUB

מספר הצהרות RETURN גדול ב-1 ממספר הצהרות GOSUB

## 8 End of file

דוח טיפול בקבצים בכונן זעיר.

## 9 STOP statement

נעשה שימוש ב-STOP לעצירת התכנית. CONTINUE תגרום להמשיך הריצה מההצהרה הבאה.

## A Invalid argument

פונקציה קיבלה ארגומנט או ערך שגוי.

## B Integer out of range

ערך עוגל לשלם הקרוב ביותר הנמצא מחוץ לתחום המותר.

## C Nonsense in BASIC

ההצהרה אינה הגיונית בביסיק בהקשר של התכנית.

## D BREAK-CONT repeats

מקש BREAK נלחץ. CONTINUE תחזור על ההצהרה בה נעצר המחשב.

## E Out of DATA

READ ניסתה לקרוא מעבר לסוף הצהרת DATA האחרונה בתכנית.

## F Invalid file name

נעשה שימוש ב-SAVE עם שם המכיל יותר מ-10 תווים.

## G No room for line

לא נותר מרחב זכרון פנוי להכנסת שורת התכנית החדשה.

## H STOP in INPUT

הוקשה בתגובה ל-INPUT או בתחילת הנתונים הנקלטים. CONTINUE תחזור על הצהרת INPUT שנעצרה.

## I FOR without NEXT

לולאת FOR NEXT לא בוצעה עקב גבולות או ערך STEP לא נכונים, (לדוגמה, FOR x=5 TO 0 without step) והצהרת NEXT המתאימה לא נמצאה.

## J Invalid I/O device

דוח טיפול בקבצים בכונן זעיר.

## K Invalid colour

הערך שהוגדר עבור INVERSE, BRIGHT, FLASH, PAPER, INK או OVER או תו הבקרה המתאים מחוץ לתחום המותר.

## L BREAK into program

מקש BREAK נלחץ. הדוח מפרט את ההצהרה האחרונה שהיה לבצע ו-CONTINUE ממשיכה מההצהרה הבאה.

## M RAMTOP no good

הערך שהוגדר עבור RAMTOP הוא גדול מדי או קטן מדי.

## N Statement lost

המחשב ניסה לקפוץ להצהרה שאינה קיימת עוד.

## O Invalid stream

דוח טיפול בקבצים בכונן זעיר.

## P FN without DEF

הצהרת FN היתה בשימוש ללא הצהרת DEF FN המתאימה.

## Q Parameter error

הצהרת FN מכילה מספר לא נכון של ערכים שיש להעביר לפונקציה, או שאחד הערכים הוא מסוג לא נכון (מחזרות במקום מספר או להיפך).

## R Tape loading error

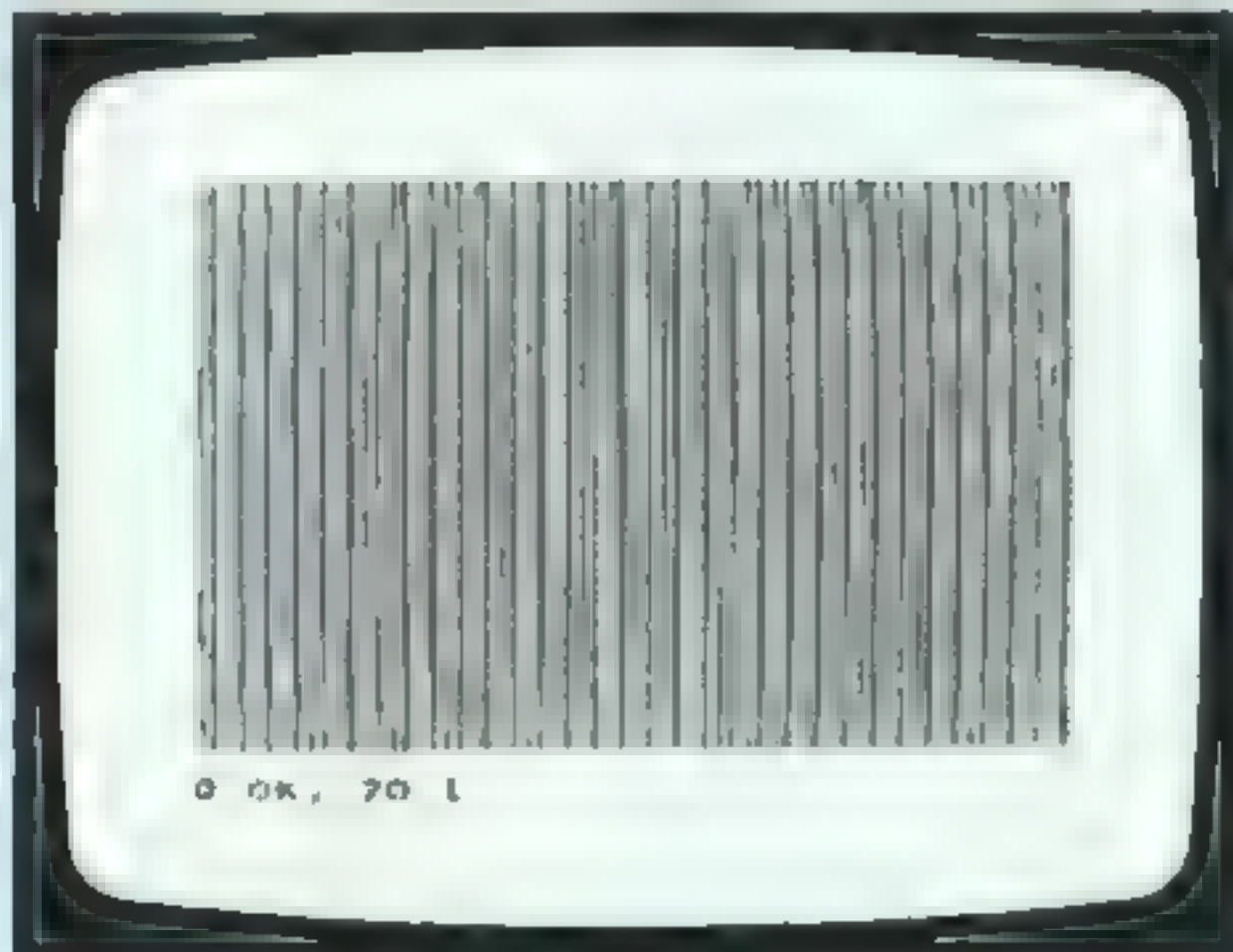
תהליך הטעינה, מיזוג או אימות נכשל.



מדריך זה. ברם, קיימים ספרים רבים המלמדים קוד מכונה בספקטרום ברמה מתקדמת. כדי לקבל מושג על מהירות הביצוע בקוד מכונה, הקש פנימה והרץ את תכנית ההדגמה הבאה.

### פסים מיידים

```
10 FOR x=0 TO 15
20 READ a: POKE 65000+x, a
30 NEXT x
40 DATA 33,255,63,1,1,24,22
50 DATA 55
60 DATA 35,11,120,177,200,114,
24,48
70 RANDOMIZE USR 65000
```



נסה לשנות 55 בשורה 50 לערך כלשהו בין 1 ל-255 וראה כיצד משתנים הפסים. אתה יכול ליצור גם פסים צבעוניים תוך שימוש בפקודת INK בתחילה. אך זוהי אינה מטרת התכנית. ראה כמה מהר נוצרת התצוגה תוך שימוש בקוד מכונה – הבצוע כמעט מייד. לבייסיק לוקח כמעט שתי שניות למלא את המסך.

תכנית זו פועלת מפני שהצהרת DATA מכילה 16 קודים המאוחסנים בכתובת 65000 והלאה בזכרון בשורות 10 עד 30. שורה 70 שולחת את הקודים ל-Z80A והתצוגה נוצרת מייד. שים לב שהקוד השמיני מבקר את רוחב הפסים.

משחקים רבים שנוצרו עבור הספקטרום כתובים בקוד מכונה ליצירת פעולה מהירה בבזק. כדי לעזור לך לכתוב קוד מכונה, קיימות תכניות הנקראות אסמבלרים. האסמבלר מספק הוראות שאתה מקיש פנימה במקום מספרים בלבד, שאותם בעצם דורש קוד המכונה. ההוראות אינן מילים באנגלית כמו מילות המפתח בבייסיק, אלא קיצורים או כינויים המייצגים את הפעולות שעל המחשב לבצע. עליך, איפוא, להבין כיצד המחשב פועל צעד אחר צעד לפני שתוכל להשתמש בשפת אסף (אסמבלר).

## מעבר לבייסיק

בייסיק היא שפת מחשב לשימוש כללי אשר פועלת היטב עבור מרבית היישומים. ברם, זוהי אינה שפת המחשב היחידה הניתנת לשימוש בספקטרום. קיימת גם תכנה המאפשרת שפות אחרות כמו FORTH, micro-PROLOG ולוגו. שפות אלה פועלות בצורה שונה לגמרי מבייסיק ופותחות אפשרויות חדשות בפני המחשב.

דווקא בגלל העובדה שבייסיק היא שפה רב-שימושית, היא יכולה להיות מסורבלת למדי ביישומים מסויימים. כמו כן היא די איטית. שפות אחרות יכולות לתת גמישות רבה יותר יחד עם פשטות בתכנות ומהירות ריצה גדולה יותר. FORTH, למשל, מאפשרת לך להגדיר מילים ולהשתמש בהן בהוראות המובנות למחשב אשר יתבצעו פי עשר מהר יותר מאשר בפקודות מקבילות בבייסיק. עם שפת micro-Prolog, המחשב יבין בטוויים פשוטים באנגלית. הוא מאחסן אותם בזכרון כבסיס לדיאלוג עם המשתמש. לוגו היא שפת מחשב שפותחה לתחום החינוכי. השפה מציעה פקודות פשוטות ביותר הניתנות לשימוש בצורה גמישה מאד. אך אם ברצונך לכתוב תכניות מהירות באמת עבור הספקטרום ZX+, עליך להבין כיצד לתכנת בקוד מכונה.

### קוד מכונה

שפת בייסיק קיימת על מנת לאפשר לך לתת הוראות למחשב בצורה הנוחה לך להבנה. היע'מ של הספקטרום – שבב Z80A רב העצמה – אינו מבין למעשה בייסיק. חלק בזכרון מכיל תכנית קבועה הנקראת מתורגמן הבייסיק אשר הופכת את הוראותיך בבייסיק לרצף של אותות קוד. קודים אלה מפעילים למעשה את ה-Z80A לביצוע הוראותיך. למתורגמן נדרש פרק זמן מסויים כדי לתרגם את הוראותיך בבייסיק לקוד Z80A או לקוד מכונה, כפי שהוא נקרא. ברם, אתה יכול לעקוף את מתורגמן הבייסיק ולשלוח ישירות קוד מכונה ל-Z80A. התכנית תתבצע במהירות רבה מאד. המחיר שעליך לשלם תמורת מהירות הבצוע הוא הזמן הנוסף שיש להשקיע בכתיבת תכנית בקוד מכונה. שלא כמו בייסיק, זוהי שפה מאד "לא ידידותית", המחייבת זמן רב ללימוד. תכנות בקוד מכונה הוא מעבר להקף של



# מילון מונחי מחשב

ישנן מילים רבות הנמצאות בשימוש בתחומי המחשב אך מופיעות גם בחיי היומיום במשמעות שונה. להלן מופיעים הסברים לכמה מילים כאלה המופיעות בספר זה, יחד עם מונחי מחשב מיוחדים. למילים המודפסות באותיות מוטות מופיעים פירושים משלהן. אם מופיעה בספר זה מילה או מונח שאינך מבין ואינו נמצא כאן, נסה לחפשו במפתח הממוין (אינדקס).

**אוגר** – יחידת זכרון קטנה הנפרדת מהזכרון הראשי. האוגרים בתוך ה'ע'צ' משמשים לביצוע תהליכי החישוב.

**אופן פעולה** – בספקטרום, זהו אחד מתוך חמישה מצבים, המכתיב אילו 'מילות מפתח' ו'זרים' ניתן להפיק מכל מקש במקלדת. בזמן התכנות, אופן הפעולה מצוין ע"י אות מהבהבת בתוך ה'סמך'.

**אופרטור** – פקודה המבצעת פעולה חשבונית או 'לוגית'.

**אמת** – כל מצב או תוצאה אשר המחשב מחליט שהם אמת או נכונים. לאמת יש ערך מספרי של 1.

**ארגומנט** – ערך הנמצא בשימוש 'פונקציה' לקבלת תוצאה מסוימת.

**בטי** – צירוף של 'קבועים', 'משתנים' ו'מילות מפתח'.

**ביסיק** – שפת המחשב אשר ספקטרום ZX+ משתמש בה, וכמוהו גם מרבית המחשבים הביתיים.

**בית** – מערך של 8 'סיביות' המייצג מספר בעל ערך הנע בין 0 ל-255. כל כתובת ב'זכרון' מכילה בית אחד.

**גלגול** – התנועה אשר מאפשרת תצוגה העולה על הגודל המוצג במסך יחיד.

**גרפיקה** – 'יצירת דימויים' כמו תמונות, טבלאות או דיאגרמות ע"י המחשב.

**דוח** – הודעה המוצגת ע"י המחשב ומתארת את פעולותיו.

**הדפס (PRINT)** – להציג תוצאות או 'גרפיקה' על המסך, או להדפיס במדפסת.

**הכנס (ENTER)** – לתת הוראה מלאה או מידע למחשב.

**הפרדה** – דרגת החרדות או הפרדת הפרטים האפשרית ב'גרפיקה' במחשב.

**הצהרה** – 'מילת מפתח' המשמשת ליצירת פקודה בתוך 'שורה' בתכנית, או הפקודה עצמה.

**זכרון** – החלק במחשב אשר מכיל את ה'תכנית' וה'ציונים' כאשר הם נדרשים, וכמו כן את הוראות ההפעלה הקבועות.

**חמרה** – המחשב עצמו וכל מכונה או התקן הקשורים בו, כמו 'רכיבים היקפיים'.

**טעינה** – הזנת 'תכנית' או 'ציונים' למחשב מתוך התקן אחסון כמו קלטת או מחסנית.

**ע'מ' (CPU)** – יחידת עיבוד מרכזית. החלק המרכזי במחשב, אשר מבצע את החישובים ומבקר את היחידות האחרות. בספקטרום ZX+ יש מעבד זעיר מסוג Z80.

**כתובת** – יחידה בודדת של ה'זכרון' בספקטרום ZX+ קיימות 65536 כתובות.

**לוגיקה** – תהליך שלפיו המחשב מחליט האם התוצאות הן נכונות או מוטעות או האם מצב הבטוי הנבדק הוא 'אמת' או 'שקר'.

**לזלאה** – קטע ב'תכנית' החוזר על עצמו פעם אחת או יותר.

**מאפיינים** – קודים המציינים את צבעי התוים.

**מחרוזת** – קבוצת 'זרים' (אחד או יותר) המובדלת במרכאות כדי לבדדה ממספרים ו'משתנים' מספריים.

**מידע** – מילים, מספרים וסימנים בכל צירוף שהוא, אשר המחשב נדרש לטפל בו.

**מילת מפתח** – הוראה למחשב בשפת בייסיק. יתכן ונדרשים גם מספר 'ערכים' לצורך הפעולה.

**מימשק** – יחידה המחברת את המחשב ו/או 'רכיבים' היקפיים יחד ומבטיחה אפשרות תקשורת ביניהם.

**מערך** – קבוצת 'ציונים' הנמצאת יחד באחד מחלקי ה'זכרון'.

**מערך תוים** – המערך המושלם של 'זרים' קבועים מראש ומספר קודי בקרה הנמצאים בשימוש המחשב.

**משתנה** – אחת או יותר יחידות 'זכרון' המייצגות 'קבוע' מסויים לשימוש ע"י המחשב. לכל משתנה ניתן שם או אות לצורך זיהויו. הספקטרום ZX+ מבחין בין 'משתנים' מספריים ו'משתני מחרוזת'.



**משתנה מחרוזת** – משתנה אשר מייצג 'מחרוזת':  
משתני מחרוזת בנויים תמיד מאות אחת וסימן דולר  
(\$).

**משתנה מספר** – משתנה אשר מייצג מספר.  
משתנים מספריים מורכבים מאות אחת או יותר.

**נתונים** – 'זיכרון' המגיע למחשב מתוך 'תכנית' או  
המוזן מבחוץ לתוך המחשב לצורך קבלת תוצאות.

**סיבית** – מצב 'זע' או 'תק' ב'קוד בינארי'. זהו שם  
קיצור לספרה בינארית.

**סמן** – המקום על פני המסך בו עתיד להיות מוצג  
התו הבא. ניתן לסמנו ע"י סימן מהבהב המציין את  
'אופן הפעולה' בו נמצא המחשב.

**עורך (EDIT)** – שינוי פרטים בתוך 'תכנית'.

**ערך** – כל מספר או מחרוזת אשר ניתן להציב בהם  
או לייצגם ע"י 'קבוע', 'משתנה' או 'בטוי'.

**פונקציה** – תהליך שבו המחשב לוקח אחד או יותר  
'ערכים' (או 'ארגומנטים') ומשתמש בהם לקבלת  
תוצאה שהיא ערך אחר.

**פיקסל** – נקודת הצבע הקטנה ביותר המופיעה על  
המסך. זהו שם מקוצר (מאנגלית) ל'תא תמונה'.

**פלט** – תוצאות המופקות ע"י המחשב.

**פקודה** – הוראה בודדת המתבצעת ע"י המחשב.

**פקודה ישירה** – מערך של הוראה אחת או יותר  
המתבצעת מיידית עם הגיעו למחשב.

**ק'** – מידת קיבול ה'זיכרון' של מחשב. 1 ק' שווה ל-1  
קילרבייט או 1024 סיביות. קיבול הזיכרון בק' שווה  
למספר הכללי של 'כתובות' בזיכרון, כשכל אחת יכולה  
לאחסן סיבית אחת. לספקטרום ZX+ יש 48 ק' 'זיכרון'  
16 ק' 'זיכרון' הנותנים בסה"כ 64 ק'.

**קבוע** – מספר או קבוצה של אות אחת או יותר, או  
כל 'ערך' אחר.

**קוד בינארי** – סוג הקוד הנמצא בשימוש המחשב.  
הקוד כולל סדרת מצבי 'זע' - 'תק' – לדוגמה,  
דחפים חשמליים במצבי 'זע' - 'תק'.

**קוד מכונה** – השפה אותה מבין הספקטרום ZX+.  
תכניות בייסיק מתורגמות לקוד מכונה בזמן הרצתן  
ע"י המחשב.

**קינזון** – ארגון 'לולאות' בתוך תכנית כך שלולאה  
אחת או יותר מתבצעות בתוך לולאה אחרת.

**קלט** – 'תכניות' ו'נתונים' המוחנים למחשב.

**ראם (RAM)** – זיכרון בגישה אקראית. חלק ב'זיכרון'  
היכול להכיל 'תכנית' ו'נתונים' וערכים משתנים  
אחרים. ידוע גם כזיכרון נדיף (לא קבוע). תוכן זיכרון  
ראם נמחק כאשר מתנתקת אספקת הזרם.  
לספקטרום ZX+ יש 58 ק' ראם.

**רום (ROM)** – זיכרון קריאה בלבד. החלק ב'זיכרון'  
אשר מכיל 'תכניות' קבועות ופקודות למחשב.  
לספקטרום ZX+ יש 16 ק' רום.

**רכיב היקפי** – כל רכיב המחובר למחשב.

**רשום** – 'שורות' בתכנית המפורטות לפי הסדר  
ברשימה על המסך או מדפסת.

**שורה** – פקודה או מערך פקודות ב'תכנית' לכל  
שורה ניתן מספר כך שהיא תתבצע בנקודה המדויקת  
בתוך סדר פעולת כל השורות.

**שמור (SAVE)** – אגירת 'תכנית' או 'נתונים' בהתקן  
אחסון כמו קלטת או תקליטון.

**שקר** – כל מצב או תוצאה אשר המחשב מחליט  
שהוא לא אמת או לא נכונה. לשקר יש ערך מספרי  
של 0.

**שדשור** – צירוף מספר 'מחרוזות' יחד ע"י הוספתן  
אחת לאחת.

**תו** – אות בודדת כלשהי, מספר (0 עד 9), סימן או  
יחידת 'גרפיקה' הניתנים לתצוגה או הדפסה.

**תחביר** – הסדר הנכון של 'צילוח' מפתח, 'קבועים',  
'משתנים', ו'בטויים' הדרושים לצורך יצירת פקודת  
בייסיק תקפה.

**תכנה** – כל 'תכנית', כולל תכניות קבועות בזיכרון  
'זיכרון' או מחסניות.

**תכנית** – קבוצת פקודות בסדר מסויים המתבצעת  
ע"י המחשב.



# מפתח ממוין (עברית)

מספרי עמודים מודגשים מתייחסים  
לאיורים ומסגרות.

## אופני פעולה 28-1

- אותיות 20; 21
- אותיות גדולות 20; 21
- מילת מפתח 20; 21
- גרפיקה 20; 21
- אופרטורים אריתמטיים 22; 22
- אחסון 44, 45
- אנטנה, שקע וכבלים 4-5
- אפקטים אקראיים 30
- אפקטים קוליים 36-7

## בייסיק 73 - 49, 18

בייסיק סינקלייר 49-73

## גובה צליל מוסיקלי 36

גלגול 8

## גרפיקה, הנפשה 34-5

- צבע 24-5
- יצירת תווים 32-3
- מילוי צורות 29; 29
- הפרדה גבוהה 26, 28-9
- הפרדה נמוכה 26-7
- צורות 30-1
- אפקטים אקראיים 30

## דוחות מסך 74

דיאגרמות עמודות 25; 25

## הגברת קול 37

הכנסת תכניות 8-9

הנפשה 5 - 34

הצהרות 22, 50

הקשה פנימה 8, 9

הרצת תכניות 8, 9

התנגשויות 34-5

## וסת מתח 43

זכרון 8-44, 43, 42; 12

## חבורים 5

- רשמקול קלטות 13
- ספק כח 5
- טלביזיה 4
- חמרה, הגדרה 12
- חשוכים 22, 23; 22-3

טון, בקרי רשמקול 12, 14, 15

טלביזיה, חבור 5

- התאמה 4

- כוונן 6; 6

- מקודד 42

טעויות, תיקון 10, 21

טעינה 13-14; 14-15

יע"מ (CPU) 43, 45; 43, 44, 48, 75

כבל שטוח 46

כונן זעיר 46, 5; 46

- תקליטונים 12, 45

- טעינה 46

כפתור RESET 5; 11, 12

לולאות 26-7, 30

מדפסות 45, 47; 45, 47

מודמים 46

מוט הגוי 45, 47

מוסיקה 36-7

מחבר קצה 43, 47, 5

מחרוזות 22

מילות מפתח 1-20, 52-73, 50, 18-19, 9

- בחירה 19, 20

ממשקים 45; 46-7

ממשק 1 ZX 45, 46-7

ממשק רדיו 4

ממשק RS-232 45; 47

מספרים 50

מערך לוגי בלתי תלוי (ULA) 42

מערך תווים 51

מפוי זכרון 48

מקלדת 18-19

מקשים 18-19; 18-19

- הפעלה 1-20; 20-1

מקש רווחים 19

מקשי מספרים 19

מעבד זעיר 280 45; 75, 43

משתנים 3, 50 - 22

משתני מערכת 48

נקודה 23, 51

נקודתיים 23, 51

נקודה-פסיק 23, 51

סוגרים 23

סימנים, חישובים 22, 50

- בחירה 19

סימן מרכאות 23, 51

סימני פיסוק 23, 51

סמלים, בחירה 20

סמן, בקרים 19

ספק כח 43, 5; 5, 4

סרטים 12, 45

טיפול 12

- סימן 14, 39

- השמעה 12

- אחסון 12

סרטי קלטות 44-45, 12

- טיפול 12

- סימן 14

- השמעה 12

- אחסון 12

עצמה, בקרי רשמקול 12, 14, 15

ערכת שורות תכנית 21

פונקציות 50

פיקסלים 28

פסיק 51, 23

פקודות 22, 50

צבע 24-5; 24-5

- קודים 24

- קודי בקרה 33

- צירופים 25

- מקשי תצוגה 19

- עירוב 32

- בדיקה 24; 6

צבע גבול 6; 24-5

צבע דיו 24-5

צבע נייר 24-5

צורות, מילוי 29, 29

קוד בינארי 44

קוד מכונה 75

קלט-פלט, נתיבים 45

ראם (זכרון בגישה אקראית) 42, 45;

42, 48

ראם, חבילות 4

ראם 16 ק' ZX 4

רום (זכרון קריאה בלבד) 43, 45, 48;

רום, מחסניות 7-46; 47, 12

רכיבים היקפיים 45, 46-7

רמקול 43

רשומים 8, 21

רשמקול קלטות, כמגבר 37; 37

- בחירה 12

חבורים 13; 13, 5

- מונים 14

- טעינת תכניות 14-16

- שמירת תכניות 38-40

- בקרת טון 14, 15, 16

- בקרת עצמה 14, 15, 16

רשת נקודות, הפרדה גבוהה 80, 28

- הפרדה נמוכה 80, 26

שבבים 3-42

שבבים לוגים 43

שגרות 30-1

שורות 8

- מחיקה 21

- עריכה 21

שורות תכנית, מחיקה 21

- עריכה 21

שמירה 38-40, 13

שנוי תכניות 9

שפעול תכניות 10

שקעים 5

שקע EAR 13, 5; 37

שקע MIC 13, 5; 37

שקע 9VDC 43, 5



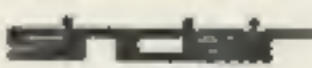
תכנית זריחה מנצנצת 11	תוים בהגדרת משתמש 32-3; 80
תכנית טבלת כפל 23	תוים, יצירה 32-3
תכנית טיוטה 29	— בחירה 20
תכנית כדור מנחר 35	תכנה 12
תכנית כוכבים 28	— טעינה 14-16; 14-16
תכנית כוכבים ופסים 11	— מוכנה לריצה 13; 12-13
תכנית מחאיקה מטורפת 10	— התאמה 12
תכנית עגולים מהבהבים 9	— סוגים 12
תכנית פוליהדרה 10	תכנות 17-40
תכנית פירמידות 31	תכניות, שנוי 9
תכנית צורות 9	— התחלת תכניות חדשות 11
תכנית צורות סימטריות 30	— תקון שגיאות 10
תכנית קשת בענן 26-7	— הכנסה 44, 8-9
תכנית רבועים 30	— טעינה 14-15; 15 — 12, 13, 14
תכנית רובוט ZX 27	— שפעול 10
תכנית שחמט 33	— הרצה 44, 8-9
תקליטונים, כונן זעיר 46; 12, 46	— שמירה 38-40, 13
	— אימות 39

עורך דוד ברני  
עורך אמנותי כטר לך  
תכנון דברה לי  
צילום טרוור מלטון  
צלם צילומי רשת וינסנט אוליבר  
ניהול עריכה אלן בקינגהם  
סדר צילום טכנוסדר בע"מ

תורגם על ידי חבר המתרגמים, תל-אביב

פורסם לראשונה ב-1984 על ידי דורלינג קינדרסליי בע"מ, רח' הנריאטה 9, לונדון, בשותפות עם סינקלייר ריסרץ' בע"מ, דרך ויליס 25, קמברידג'.  
זכות יוצרים © 1984 ע"י סינקלייר ריסרץ' בע"מ ודורלינג קינדרסליי בע"מ, לונדון.

איורים — זכות יוצרים © 1984 ע"י דורלינג קינדרסליי בע"מ, לונדון.

  
סינקלייר ZX Spectrum, ZX Microdrive, ZX Interface הם סימנים מסחריים של סינקלייר ריסרץ' בע"מ.

כל הזכויות שמורות. אין לשכפל, לאחסן במערכת איחזור או להעביר בכל צורה או בכל אמצעי, אם זה אלקטרוני, מכני, צילום, הקלטה או בכל דרך אחרת, חלק כלשהו מפרסום זה ללא ההסכמה המוקדמת בכתב של בעלי זכויות היוצרים.











## תוכנת ספקטרום

כל מגוון התוכנה עבור מחשבי ספקטרום (כולל כל הנושאים הקיימים כיום) תואם בשלמותו למחשב החדש ספקטרום ZX+.

ՁԵՐԿՈՒ ԼՈՒՈՒ, ԵՆՈՒ

ԷԹԱՆԸ ԱՅՈՒՆ

ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹԱԳՐԱԴԱՐԱՆ